# Inference of Gaussian graphical models and ordinary differential equations

Vujacic, Ivan

university of
groningen

# Inference of Gaussian graphical models and ordinary differential equations

**PhD Thesis**

to obtain the degree of PhD at the
University of Groningen
on the authority of the
Rector Magnificus Prof. E. Sterken
and in accordance with
the decision by the College of Deans.

This thesis will be defended in public on

1 July 2014 at 11:00 hours

by

**Ivan Vujačić**

born on 6 March 1983
in Podgorica, Montenegro

**Supervisor**
Prof. E.C. Wit

**Assessment committee**
Prof. M. Girolami
Prof. C. Klaassen
Prof. E.R. van den Heuvel

*To my parents*

***Nikola*** *and* ***Vjera***

# Acknowledgements

First of all, I would like to thank my supervisor Professor Ernst Wit for his help and guidance.

I am grateful to Marija Kovačević for her encouragement and help while applying for the JoinEU-SEE scholarship, which I received for my PhD position. I thank the JoinEU-SEE project for granting me the scholarship and Ernst for providing the rest of the funding.

This thesis is richer as a result of collaborations with Antonino Abbruzzo, Itai Dattner and Javier Gonzaléz.

I thank the professors from the assessment committee Mark Girolami, Chris Klaassen and Edwin van den Heuvel for reading my thesis. Also, I thank my office mate Abdolreza Mohammadi for reading the first three chapters.

My thanks go to Maurits Silvis for translating the summary into Dutch. Again, thanks to Ernst for improving the translation.

I am grateful to everyone who helped me somehow during this work.

Finally, special thanks go to my family: my father Nikola, my mother Vjera, my brother Andrej and my sisters Tijana and Marija for supporting me throughout the years of work on this thesis. Also thanks to Tijana's family: my brother-in-law Novak, my nephew Uroš and my niece Simona.

Ivan Vujačić, May 2014

*"Bachatu, Bachatu..."*

***Antony Santos,***
***El Mayimbe***

# Contents

**Contents** vii

**List of Figures** xi

**List of Tables** xiii

**Notation** xv

**Symbols** xxii

**Chapter 1: Introduction** 1

1.1 Motivation . . . 1
1.2 Estimating Gaussian graphical models . . . 2
1.2.1 Introduction . . . 2
1.2.2 References . . . 3
1.3 Estimating parameters in ordinary differential equations . . . 5
1.3.1 Introduction . . . 5
1.3.2 References . . . 8
1.4 Our work and contribution . . . 8
1.5 Outline of the thesis . . . 10

**Chapter 2: Model selection in Gaussian graphical models** 11

2.1 Undirected graphical models . . . 12
2.2 Gaussian graphical models . . . 14
2.3 Kullback-Leibler information . . . 15
2.4 Penalized likelihood estimation in Gaussian graphical model . . . 18
2.4.1 Estimation . . . 18
2.4.2 Model selection . . . 20

2.4.3 Measuring the goodness of a model . . . . . 22
2.5 Summary . . . . . 23

**Chapter 3: Estimating the KL loss in Gaussian graphical models 25**
3.1 Prediction VS graph structure . . . . . 26
3.2 The GIC and the KLCV as estimators of the KL loss . . . . . 27
3.3 Derivation of the GIC and the KLCV . . . . . 28
3.3.1 Derivation of the GIC for the maximum likelihood estimator . . 28
3.3.2 Derivation of the KLCV for the maximum likelihood estimator . 30
3.3.3 Extension of the GIC and the KLCV for the maximum penalized likelihood estimator . . . . . 31
3.4 Implementation . . . . . 33
3.5 Simulation study . . . . . 34
3.6 Using the KLCV and the GIC for graph estimation . . . . . 35
3.7 Summary . . . . . 38

Appendices
3.A Proof of Theorem 3.1 . . . . . 38
3.B Matrix differential calculus . . . . . 43
3.C Calculation of the derivatives . . . . . 45
3.D Derivation of the expression for $\mathbf{Q}$ . . . . . 45
3.E Proof of Lemma 3.1 . . . . . 46
3.F Calculation of the algorithmic complexity . . . . . 46

**Chapter 4: Time course window estimator for ordinary differential equations linear in the parameters 49**
4.1 Introduction . . . . . 50
4.2 Time-course window estimator . . . . . 51
4.3 Simulation examples . . . . . 55
4.3.1 Empirical validation of $\sqrt{n}$-consistency . . . . . 56
4.3.2 Comparing different error distributions . . . . . 57
4.3.3 Window-based estimator as initial estimate . . . . . 60
4.4 Computational complexity . . . . . 61
4.5 Real data example . . . . . 62
4.6 Discussion . . . . . 63
4.7 Summary . . . . . 65

Appendices
4.A Proof of Theorem 1 . . . 65
4.B Auxiliary results . . . 69
4.C Calculation of the algorithmic complexity . . . 70
4.D Calculation of the integrals . . . 72

**Chapter 5: RKHS approach to estimating parameters in ordinary differential equations 77**
5.1 Preliminaries . . . 77
5.1.1 Reproducing kernel Hilbert spaces . . . 77
5.1.2 Green's function and reproducing kernel Hilbert spaces . . . 80
5.2 Explicit ODEs . . . 82
5.3 RKHS based penalized log-likelihood . . . 84
5.4 Approximate ODEs inference . . . 86
5.4.1 Model selection . . . 87
5.5 Examples using synthetically generated data . . . 88
5.5.1 Explicit ODEs versus regularization approach . . . 88
5.5.2 Comparison with the MLE . . . 89
5.5.3 Influence of the sample size on the estimation . . . 90
5.5.4 Comparison with generalized profiling procedure . . . 94
5.6 Real example: Reconstruction of Transcription Factor activities in Streptomyces coelicolor . . . 94
5.7 Summary . . . 96

Appendices
5.A Proof of Proposition 1 . . . 97
5.B Derivation of the AIC for the maximum penalized likelihood estimator 98

**Chapter 6: Inferring latent gene regulatory network kinetics 101**
6.1 System and methods . . . 102
6.1.1 Modelling transcriptional GRN with ODE models . . . 102
6.1.2 GRN with one TF: single input motif . . . 103
6.1.3 Noise model . . . 104
6.1.4 Penalized log-likelihood of a GRN with one TF . . . 104
6.1.5 Parameter estimation . . . 106
6.1.6 Model selection . . . 106

6.1.7 Confidence intervals . . . 106
6.2 Algorithm . . . 107
6.2.1 Augmented data formulation . . . 107
6.2.2 E-step . . . 108
6.2.3 M-step . . . 108
6.2.4 EM algorithm . . . 109
6.3 SOS repair system in Escherichia coli . . . 109
6.3.1 The data set and the goal . . . 110
6.3.2 Estimation process . . . 110
6.3.3 Reconstruction of the LexA activity . . . 111
6.3.4 Inferred kinetics profiles . . . 112
6.3.5 Estimated kinetic parameters and interpretation . . . 112
6.4 Summary . . . 113

Appendices
6.A Notation . . . 113
6.B Proof of the expectation step of the EM algorithm . . . 115
6.C Proof of the maximization step of the EM algorithm . . . 116

**Conclusion** **117**

**Summary** **119**

**Samenvatting** **121**

**References** **123**

# List of Figures


3.1 Hub graphs with $p = 40$ and $p = 100$ nodes used in the simulation study. 35

3.2 Simulations results for hub graph with $p = 100$ nodes. Average performance in terms of $F_1$ score of different estimators for different sample size $n$ is shown. The results are based on 100 simulated data sets. . . . 39

4.1 The Lotka-Volterra system. The solid lines correspond to the states $x_1$ and $x_2$ as given by the model (4.11) with $\theta_1 = 0.1$, $\theta_2 = 0.9$, $\theta_3 = 0.9$, $\theta_4 = 0.5$. The bold step functions correspond to the window smoothers of $x_1$ and $x_2$. The data, represented by the circles in the figure, consist of 100 equidistant observations on the interval $[0, 49.9]$. . . . . 58

4.2 *FhNdata*. The data are represented by the circles. The curves are obtained by solving the FitzHugh-Nagumo system for the parameter $(\widehat{a}_1, \widehat{b}_1, \widehat{c}_1)^\top = (0.017, -0.007, 0.160)^\top$ (dashed lines) and $(\widehat{a}_2, \widehat{b}_2, \widehat{c}_2)^\top = (0.318, -0.140, 3.003)^\top$ (solid lines) and initial condition $(\widehat{\xi}_1, \widehat{\xi}_2)^\top = (0.051, 0.569)^\top$. . . . . 61

4.3 England Wales measles data. Upper panel: The solid line is the solution $S(\cdot)$ based on the parameter estimate $\beta$ and the dashed line is the estimated $S(\cdot)$. Lower panel: The data are given by the circles, the solid line is the solution based on the parameter estimate $\beta$ and the stepwise solid line is the window estimator. . . . . 64

5.1 The results obtained for the differential equation $x'(t) = \theta x(t)$. . . . . 89

5.2 The results obtained for the Lotka-Volterra equations. . . . . 89

5.3 Solution of the FHN model and generated data points in one run of the experiment ($n = 100$). . . . . 91

5.4 95% confidence intervals for the parameters $a$, $b$, $c$ and $\sigma^2$ of the FHN equations for different sample sizes. Horizontal grey lines represent the true values of the parameters. The results are obtained using 50 runs of the experiment. . . . . 92
5.5 Box-plots for the absolute errors $|\widehat{a}_i - a|$, $|\widehat{b}_i - b|$ and $|\widehat{c}_i - c|$ in the estimation of the parameters of the FHN equations. The results are obtained using the generalized profiling and the maximum penalized likelihood approach (MPLE-rkhs) proposed in this work for $n = 50$. . . 93
5.6 Reconstructed genes profiles and master activator cdaR. . . . . 95

6.1 Single Input Motif (SIM) of a gene regulatory network with one transcription factor. . . . . 102
6.2 Reconstruction of the activity of the master repressor LexA scaled between 0 and 1. The smoothed LexA profile is obtained using a cubic spline. Time is given in minutes. . . . . 111
6.3 Data and reconstructed profiles of two genes which represent the two expression patterns found in the database. Raw data are represented by empty points. Dense points represent the values of the estimated profiles in the 6 observed and 20 hidden points of each gene. . . . . 112

# List of Tables

1.1 List of methods for choosing the regularization parameter in Gaussian graphical models. . . . . 4

1.2 List of general methods for solving the inverse problem of ODEs starting from the seminal paper Ramsay et al. (2007). . . . . 7

3.1 Simulation results for hub graph with $p = 40$ nodes. Performance in terms of Kullback-Leibler loss of different estimators for different sample size $n$ is shown. The results are based on 100 simulated data sets. Standard errors are shown in brackets. The best result is boldfaced and the second best is underlined. . . . . 36

3.2 Simulation results for hub graph with $p = 40$ nodes. Performance in terms of Kullback-Leibler loss of different estimators for different sample size $n$ is showed. The results are based on 100 simulated data sets. Standard errors are shown in brackets. The best result is boldfaced and the second best is underlined. . . . . 37

4.1 The empirical mean and standard deviation (in parentheses) of the window estimator of the parameters of the HIV dynamics model with Gaussian error. Results are based on 500 simulations. . . . . 55

4.2 Empirical mean and standard deviation (in parentheses) of window estimator of parameters and initial values of Lotka-Volterra system with Gaussian and Laplace error. The results are based on 500 simulations. . 59

4.3 Fitz-Hugh Nagumo system. The data are obtained from the R package `CollocInfer` and comprise 41 equally spaced observations on the interval $[0, 20]$. The window estimator does not require an initial guess (first row) and as such it can be used as initial guess for the generalized profiling method (second row). Estimates obtained by the generalized profiling method with initial guesses $10^{i-1}u$, $i = 1, \ldots, 5$, where $u = (1, 1, 1)^\top$, are shown in other rows. The best results are boldfaced. Comparison in terms of running time is only between the two best, boldfaced estimates. . . . . 62

5.1 Mean square error for the inferred parameters in the Lotka-Volterra model. Standard deviations shown in parenthesis. The true value of the parameters are fixed to $\theta_1 = 0.2$, $\beta_1 = 0.35$ $\theta_2 = 0.7$, $\beta_2 = 0.40$. The best result for each comparison is boldfaced. . . . . 91

5.2 Average, maximum and minimum errors for the estimation of the parameters of the FHN system achieved by the generalized profiling and the maximum penalized likelihood approach (MPLE-RKHS) proposed in this work. Two different sample sizes $(50, 300)$ are used for the comparison. The best result for each comparison is boldfaced. . . . . 94

6.1 Parameter estimates and confidence intervals for the 14 genes of the Ecoli-SOS system. Above, genes recN and umuC whose expression does not decline after minute 20. Below, the 12 remaining genes of the database which decline after minute 20 sorted by the ratio $r_k$. 95 % confidence intervals are calculated using a parametric bootstrap. . . . . 114

# Notation

In order to distinguish between one-dimensional and multidimensional objects we use boldface symbols. Scalars are not boldfaced while vectors and matrices are. To differentiate between vectors and matrices we use italic symbols. A vector is denoted by an italic boldface symbol whereas a matrix is denoted by an upright boldface one. Here follows a more detailed list of math fonts and conventions used in this thesis.

- *Scalar variables* are denoted by lower case italics or Greek symbols (e.g. $t$, $\lambda$ ).
- *Functionals* and *scalar functions* are denoted by lower- or upper case italics or lower case Greek symbols (e.g. $\delta_x$, $l$, $L$, $\psi$ ).
- *Vectors* are denoted by bold italics or bold Greek (e.g. $\boldsymbol{y}$, $\boldsymbol{Y}$, $\boldsymbol{\mu}$ ).
- *Vector functions* are denoted by bold italics or bold Greek (e.g. $\boldsymbol{f}$).
- *Matrices* are denoted by bold upper case Roman or bold upper case Greek (e.g. $\mathbf{A}$, $\boldsymbol{\Theta}$).
- *Matrix functions* are denoted by bold upper- or lower case Roman (e.g. $\mathbf{F}$, $\mathbf{g}$).
- *Operators* are denoted by upper case Roman (e.g. P, G).
- *The derivative* is upright sans serif $\mathsf{D}$, *differential* is lower case Roman d and the symbol $\partial$ is used for *partial differential*.
- Symbol $\varepsilon$ is used for stochastic noise and $\epsilon$ for positive real number.
- Sub/superscripts that are variables are in italics or Greek, while those that are labels are Roman (e.g. $\boldsymbol{y}_k$, $l_\lambda$, $\mathrm{BIC}_{\mathrm{KLCV}}$ ).
- Brackets are arranged in the order $[\{()\}]$.

# Symbols

### Roman Symbols

| | |
|---|---|
| $(\mathbf{X})_{ij}$ | $(i,j)$th entry of the matrix $\mathbf{X} = (x_{ij})$, i.e. $(\mathbf{X})_{ij} = x_{ij}$ |
| $\mathbf{A} \circ \mathbf{B}$ | Schur (Hadamard) product of two matrices: $(\mathbf{A} \circ \mathbf{B})_{ij} = (\mathbf{A})_{ij}(\mathbf{B})_{ij}$ |
| $\mathbf{A} \otimes \mathbf{B}$ | Kronecker product of two matrices |
| $\mathbf{A}^\top$ | the transpose of a matrix $\mathbf{A}$ |
| $\mathbf{D}_n$ | matrix of a difference operator |
| df | degrees of freedom |
| $\mathrm{df}(\lambda)$ | degrees of freedom of the MPLE estimator $\widehat{\mathbf{\Theta}}_\lambda$ of the precision matrix; it is equal to the number of nonzero elements in the upper diagonal of $\widehat{\mathbf{\Theta}}_\lambda$ |
| $\mathrm{E}_q$ | expectation with respect to a distribution $q$ |
| $\mathbf{G}$ | Green's matrix, discrete analogue of Green's function |
| $\mathbf{I}_p$ | identity matrix of order $p$ |
| $\mathbf{I}_\lambda$ | indicator matrix whose entry is 1 if the corresponding entry in the precision matrix $\widehat{\mathbf{\Theta}}_\lambda$ is nonzero, and zero if the corresponding entry in the precision matrix is zero |
| $\mathbf{K}$ | $\mathbf{K} \stackrel{\text{def}}{=} (k(x_i, x_j))_{i,j}$ is kernel (or Gram) matrix; $x_i, x_j \in X$, where $k$ is a kernel on $X$ |
| $\mathbf{K}_p$ | commutation matrix $\mathbf{K}_p$ of dimension $p^2 \times p^2$; it has the property: $\mathbf{K}_p \mathrm{vec}\mathbf{A} = \mathrm{vec}\mathbf{A}^\top$ for any $p \times p$ matrix $\mathbf{A}$ |

| | |
|---|---|
| $\langle f, g \rangle_{\mathcal{H}}$ | inner product in Hilbert space $\mathcal{H}$ |
| $\mathcal{D}$ | data set |
| $\mathcal{G}$ | graph $(V, E)$ |
| $\mathbf{M}_p$ | the matrix defined as $\mathbf{M}_p = (\mathbf{I}_{p^2} + \mathbf{K}_p)/2$ |
| $\mathcal{N}_d(\boldsymbol{y}; \boldsymbol{\mu}, \boldsymbol{\Theta}^{-1})$ | density function of variable $\boldsymbol{y}$ of a Gaussian distribution with mean $\boldsymbol{\mu}$ and precision matrix $\boldsymbol{\Theta}$ |
| $\mathcal{N}_p(\boldsymbol{\mu}, \boldsymbol{\Theta}^{-1})$ | $p$-dimensional Gaussian distribution with mean vector $\boldsymbol{\mu}$ and precision matrix $\boldsymbol{\Theta}$ |
| $\|\boldsymbol{x}\|_{\mathbf{A}}$ | norm induced by a symmetric positive definite matrix $\mathbf{A}$, $\|\boldsymbol{x}\|_{\mathbf{A}} = \boldsymbol{x}^\top \mathbf{A} \boldsymbol{x}$ |
| $\|f\|_{\mathcal{H}}$ | norm in Hilbert space $\mathcal{H}$ |
| $\mathbf{O}_p$ | zero matrix of order $p$ |
| $\mathbf{0}_d$ | $d$-dimensional zero vector |
| $\mathbb{R}$ | the set of real numbers |
| $\mathbf{S}$ | empirical covariance matrix, $\mathbf{S} = \sum_{k=1}^{n} \boldsymbol{y}_k \boldsymbol{y}_k^\top / n$ |
| $\mathbf{S}^{(-k)}$ | in GACV, this is the sample covariance matrix based on the data with the $k$th observation excluded |
| $\mathbf{S}^{(k)}$ | in cross-validation, this is the sample covariance matrix based on the $k$th partition of the data |
| $\mathbf{S}_k$ | empirical covariance matrix of $k$th observation, $\mathbf{S}_k = \boldsymbol{y}_k \boldsymbol{y}_k^\top$ |
| $\mathbf{S}_{\lambda,k}$ | influence matrix for gene $k$ |
| $\mathsf{D}f, \mathsf{D}\boldsymbol{f}, \mathsf{D}\mathbf{F}$ | the derivative of scalar, vector and matrix function, respectively |
| $\boldsymbol{t}$ | vector $(t_1, \ldots, t_n)^\top$ of time points |
| $\mathrm{diag}(d_1, \ldots, d_n)$ | diagonal matrix with the diagonal equal to the vector $(d_1, \ldots, d_n)^\top$ |
| $\mathrm{tr}(\mathbf{A})$ | trace of (square) matrix $\mathbf{A}$ |

| | |
|---|---|
| vec | the vectorization operator which transforms a matrix into a column vector obtained by stacking the columns of the matrix on top of one another |
| $\boldsymbol{x}(t)$ | vector $(x_1(t), \ldots, x_d(t))^\top$, where $\boldsymbol{x} = (x_1, \ldots, x_d)^\top$ |
| $\boldsymbol{Y}_A$ | random vector $(Y_i)_{i \in A}$, where $A \subset \{1, 2, ..., p\}$ |
| $E$ | the set of edges of a graph $\mathcal{G} = (V, E)$ |
| $G$ | Green's function |
| $k$ | kernel function |
| $L$, $L_n$ | loss functions |
| $l(\boldsymbol{\theta}\|\mathcal{D}_S)$ | the log-likelihood function based on a part of the data $\mathcal{D}_S \subset \mathcal{D}$; $\mathcal{D}$ is the entire data set |
| $l(\boldsymbol{\theta})$ | the log-likelihood function based on the entire data $\mathcal{D}$, i.e. $l(\boldsymbol{\theta}) = l(\boldsymbol{\theta}\|\mathcal{D})$ |
| $L_2$ | space of square Lebesgue integrable functions |
| $l_k(\boldsymbol{\theta})$ | the log-likelihood function based on the $k$th observation $\boldsymbol{y}_k$, i.e. $l_k(\boldsymbol{\theta}) = l(\boldsymbol{\theta}\|\boldsymbol{y}_k)$ |
| $L_\lambda$ | regularized (penalized) loss function |
| $l_\lambda$ | penalized log-likelihood function |
| $O(\cdot)$ | big O; for vector valued functions $f$ and $g$ defined on a subset of $\mathbb{R}$, we write $f(x) = O(g(x))$ as $x \to \infty$ if there exists a positive real number $M$ and a real number $x_0$ such that $\|f(x)\| \leq M\|g(x)\|$ for all $x \geq x_0$ |
| $O_p(\cdot)$ | stochastic big O; for a random vector $\boldsymbol{Y}_n$ and positive deterministic sequence $a_n \to 0$ we write $\boldsymbol{Y}_n = O_p(a_n)$ if $a_n^{-1}\boldsymbol{Y}_n$ is bounded in probability |
| $x(\boldsymbol{t})$ | vector $(x(t_1), \ldots, x(t_n))^\top$, where $x : [0, T] \to \mathbb{R}$ |
| $X \perp\!\!\!\perp Y \| Z$ | $X$ is conditionally independent of $Y$ given $Z$ |

| | |
|---|---|
| $\mathcal{H}$ | Hilbert space or Reproducing Kernel Hilbert Space |
| $\mathcal{H}_{\text{pre}}$ | inner product space spanned by finite linear combinations of functions $k(\cdot, x_i)$, where $x_i \in X$ and $k$ is a kernel on $X$ |
| $\text{cl}(i)$ | the closure of $\{i\}$ for a node $i$ |
| $\text{nei}(i)$ | the set of neighbours of a node $i$ |
| V | the set of nodes of a graph $\mathcal{G} = (V, E)$ |

**Greek Symbols**

| | |
|---|---|
| $\chi^2_d$ | the chi-squared distribution with $d$ degrees of freedom |
| $\delta_x$ | depending on context, either Dirac's functional or Dirac's function |
| $\lambda$ | regularization (penalization) parameter |
| $\Omega$ | the sample space or a regularization functional |
| $\Phi$ | feature map |
| $\boldsymbol{\Theta}$ | precision matrix |
| $\widehat{\boldsymbol{\Theta}}_\lambda$ | maximum penalized likelihood estimator of the precision matrix with regularization parameter $\lambda$ |
| $I$ | the indicator function |

**Other Symbols**

| | |
|---|---|
| $\stackrel{\text{def}}{=}$ | is equal by definition to |
| CdaR | a particular transcription factor in Streptomyces coelicolor bacterium |
| LexA | a particular transcriptional factor (repressor) in Escherichia coli bacterium |
| p53 | tumour repressor transcription factor |
| SCO3235 | a particular gene in Streptomyces coelicolor bacterium |
| SOS response | response to DNA damage |

### Acronyms / Abbreviations

| | |
|---|---|
| $F_1$ | score defined as $F_1 = 2TP/(2TP + FN + FP)$ |
| AIC | Akaike information criterion |
| BIC | Bayesian information criterion |
| CV | cross-validation |
| EBIC | extended Bayesian information criterion |
| EM | expectation maximization |
| FHN | FitzHugh-Nagumo |
| FN | false negatives |
| FP | false positives |
| GACV | generalized approximate cross-validation |
| GGM | Gaussian graphical model |
| GIC | generalized information criterion |
| GRN | gene regulatory network |
| KL | Kullback-Leibler information |
| KLCV | Kullback-Leibler cross-validation |
| LOOCV | leave-one-out cross-validation |
| MLE | maximum likelihood estimator |
| MM | Michaelis Menten |
| MPLE | maximum penalized likelihood estimator |
| MRF | Markov random field |
| mRNA | messenger Ribonucleic acid |
| ODE | ordinary differential equation |

| | |
|---|---|
| RKHS | reproducing kernel Hilbert space |
| SIM | single input motif |
| StARS | stability approach to regularization selection |
| TF | transcription factor |
| TN | true negatives |
| TP | true positives |

# Chapter 1

# Introduction

## 1.1 Motivation

On a cellular level various biological processes are continually taking place. They involve the interaction of different molecules and this *interaction* exhibits various kinds of *dynamics*. The key words here are interaction and dynamics. These can be modelled using the mathematical notions of *graph* and *ordinary differential equation*. The topic of this thesis is the statistical treatment of these mathematical concepts.

Graphs or networks are used to model interactions; an example is *gene regulatory networks* (GRNs), which are complex systems made up of genes, proteins and other molecules. It is of great interest to biologists to discover the structure of the graph that represents a particular GRN. The problem is that there are thousands of genes and data are *sparse*; i.e. we are dealing with huge networks but with few data to provide us with information about them. Fortunately, like the data, GRNs are also *sparse* in the sense that only a few elements interact with each other. The sparsity assumption can be incorporated into statistical methods. One statistical approach that incorporates sparsity into the classical statistical methodology is *penalized Gaussian graphical models* (GGMs). This is the first topic to be treated in this thesis. To be more specific, we treat the topic of model selection, i.e. how to select an appropriate GGM.

Once the graph structure of a GRN is found, it is of interest to understand the dynamics of this network. The dynamics are usually modelled by ordinary differential equations (ODEs). The equations used for modelling, generally, contain parameters that are unknown, since they depend on the network. These parameters can only be estimated from the data. This is the second topic that we treat in this thesis. More

specifically, the main problem here is computational since using classical statistical methodology requires repeated solving of ODEs. In this thesis, however, we present estimation methods that do not require using numerical solvers.

## 1.2 Estimating Gaussian graphical models

### 1.2.1 Introduction

A *graphical model* is composed of nodes that represent random variables and edges that represent conditional dependencies between variables. The *Gaussian graphical model* (GGM) is a graphical model in which the nodes are jointly normally distributed. In GGMs conditional dependencies are summarized in the inverse covariance matrix, called the *precision matrix*. Non-zero elements in the precision matrix correspond to conditionally dependent variables. Therefore the main goal in GGM is to estimate the precision matrix.

**The maximum likelihood estimator (MLE): The saturated model**

Maximum likelihood estimation is a general method for estimating the parameters of the statistical model by maximizing the log-likelihood of the data. In the case of a GGM with $p$ nodes and a data set $\mathcal{D} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\}$ of $p$-dimensional observations, the scaled log-likelihood up to an additive constant is equal to

$$\frac{2}{n} l(\boldsymbol{\Theta}) = \log |\boldsymbol{\Theta}| - \operatorname{tr}(\boldsymbol{\Theta}\mathbf{S}), \tag{1.1}$$

where $\boldsymbol{\Theta}$ is the $p \times p$ precision matrix and $\mathbf{S} = \sum_{k=1}^{n} \boldsymbol{y}_k \boldsymbol{y}_k^\top / n$ is the $p \times p$ empirical covariance matrix. The maximizer of (1.1), which almost surely exists when $n > p$, is called the *maximum likelihood estimator* (MLE) and has the form $\widehat{\boldsymbol{\Theta}} = \mathbf{S}^{-1}$. The graph that corresponds to the MLE is the full graph. This is because the maximum likelihood estimator of any entry of the precision matrix, as a continuous random variable, takes a zero value with probability zero. This model does not assume any conditional independence relations between variables and is called the *saturated model*.

**The maximum likelihood estimator under a given graphical model**

Since the MLE does not yield any zeros in the precision matrix, sparsity can be achieved by setting some of the elements of the precision matrix to zero while estimating the rest with the maximum likelihood method (Dempster, 1972). This means that we assume certain conditional independence relations, i.e. a certain graph

structure. Since there are many possible graphs, we deal with the problem of model selection: i.e. from the family of estimated precision matrices, how do we choose the precision matrix with "the right" pattern of zeros? This approach is problematic since parameter estimation and model selection are done separately, which leads to instability of the estimator (Breiman, 1996). Furthermore, when $p > n$ there is the fundamental problem of the existence of MLE (Lauritzen, 1996, p.148) and when $p$ is large there are computational problems.

**Maximum penalized likelihood estimation (MPLE)**

Setting zeros in the precision matrix can be done automatically by penalizing the coefficients of the precision matrix (Yuan and Lin, 2007). This idea was first used in regression (Tibshirani, 1996). This is achieved by maximizing the *penalized log-likelihood function* c.f. (1.1)

$$l_{\boldsymbol{\lambda}}(\boldsymbol{\Theta}) = \log|\boldsymbol{\Theta}| - \operatorname{tr}(\boldsymbol{\Theta}\mathbf{S}) - \sum_{i=1}^{p}\sum_{j=1}^{p} p_{\lambda_{ij}}(|\theta_{ij}|), \tag{1.2}$$

where $\boldsymbol{\lambda} = (\lambda_{11}, \ldots, \lambda_{pp})^\top$ are the *regularization parameters* (*tuning or penalization parameters*) and typically $\lambda_{11} = \ldots = \lambda_{pp} = \lambda$. The penalty function is chosen in a way that enforces sparsity in the estimator of the precision matrix. This approach has the advantage that estimation and model selection are done simultaneously. It can be used when $n \leq p$, because the maximizer of (1.2) exists for a particular choice of penalty. An important issue is determining the amount of penalization, which is controlled by $\boldsymbol{\lambda}$. We treat this problem in Chapter 3 after a more in depth review of GGMs in Chapter 2. The list of existing methods is given in Table 1.1.

### 1.2.2 References

Dempster (1972), Lauritzen (1996), Edwards (2000), Rue and Held (2005), Meinshausen and Bühlmann (2006), Yuan and Lin (2007), Rothman et al. (2008), Friedman et al. (2008), Banerjee et al. (2008), Whittaker (2009), Fried and Vogel (2010), Lam and Fan (2009), Fan et al. (2009), Liu et al. (2010), Foygel and Drton (2010), Menéndez et al. (2010), Bühlmann and Van De Geer (2011), Schmidt (2010), Ravikumar et al. (2011), Cai et al. (2011), Lian (2011), Gao et al. (2012), Fitch (2012), Chen (2012), Liu et al. (2012), Fellinghauer et al. (2013), Voorman et al. (2013).

| Authors | Method | Type | Goal | Theoretical properties | R package |
|---|---|---|---|---|---|
| Akaike (1973) | AIC | closed-form criterion | prediction | not established | N.A. |
| Stone (1974) | $K$-CV | computational | prediction | not established | huge |
| Yuan and Lin (2007) | BIC | closed-form criterion | graph identification | graph selection consistent with SCAD and Adaptive LASSO penalties $n \to \infty$; $p$ fixed | huge |
| Foygel and Drton (2010) | EBIC | closed-form criterion | graph identification | graph selection consistent with SCAD penalty $n \to \infty$; $p \to \infty$ | huge |
| Liu et al. (2010) | StARS | computational | graph identification | partially sparsistent; $n \to \infty$; $p \to \infty$ | huge |
| Lian (2011) | GACV | closed-form criterion | prediction | not established | N.A. |

Table 1.1 List of methods for choosing the regularization parameter in Gaussian graphical models.

## 1.3 Estimating parameters in ordinary differential equations

### 1.3.1 Introduction

A *system of ordinary differential equations* is the set of equations that relate the values of the unknown functions of one variable and their derivatives of various orders. In this thesis we consider the systems of the form

$$\begin{cases} \boldsymbol{x}'(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t; \boldsymbol{\theta}), \ t \in [0, T], \\ \boldsymbol{x}(0) = \boldsymbol{\xi}, \end{cases} \tag{1.3}$$

where $\boldsymbol{x}(t)$ takes values in $\mathbb{R}^d$, $\boldsymbol{\xi}$ in $\Xi \subset \mathbb{R}^d$, and $\boldsymbol{\theta}$ in $\Theta \subset \mathbb{R}^p$ and $\boldsymbol{f}$ and $\boldsymbol{u}$ are known functions. Function $\boldsymbol{x}(t)$ is called the *state* and $\boldsymbol{f}$ is called the *vector field*. If the function $\boldsymbol{f}$ on the right hand side of (1.3) depends only on $\boldsymbol{x}(t)$ and $\boldsymbol{\theta}$, then the system is called *autonomous*. Given the values of $\boldsymbol{\xi}$ and $\boldsymbol{\theta}$, we denote the solution of (1.3) by $\boldsymbol{x}(t; \boldsymbol{\theta}, \boldsymbol{\xi})$. Let us assume that a process is modelled by ODEs (1.3). The parameters $\boldsymbol{\xi}$ and $\boldsymbol{\theta}$ are not known and the aim is to estimate them from noisy observations $\boldsymbol{y}(t_i)$, $i = 1, \ldots, n$ of the true state $\boldsymbol{x}(t; \boldsymbol{\theta}, \boldsymbol{\xi})$ at certain time points $t_i \in [0, T]$, $i = 1, \ldots, n$:

$$\boldsymbol{y}(t_i) = \boldsymbol{x}(t_i; \boldsymbol{\theta}, \boldsymbol{\xi}) + \boldsymbol{\varepsilon}(t_i), \quad i = 1, \ldots, n.$$

This problem is called the *inverse problem of ordinary differential equations*.

Although ordinary differential equations are ubiquitous in science and engineering, the inverse problem of ODEs is challenging. The principal notion in statistics is that of the log-likelihood function, and the main problem in estimation of ODEs lies in the fact that evaluation of the log-likelihood function requires knowledge of the solution of the ODEs. Since, in general, ODEs do not have a closed-form solution it follows that the log-likelihood cannot be obtained explicitly. Indeed, for the sake of simplicity let us assume that we have only one equation and that the noise is Gaussian with zero mean and variance $\sigma^2$. Then the log-likelihood function is up to an additive constant

$$l(\boldsymbol{\theta}, \xi) = -\frac{1}{2\sigma^2} \sum_{i=1}^{n} \{y(t_i) - x(t_i; \boldsymbol{\theta}, \xi)\}^2, \tag{1.4}$$

which involves the solution $x(t; \boldsymbol{\theta}, \xi)$ of the ODEs (1.3).

**Classical methods coupled with numerical solvers**

An obvious approach to circumvent this problem is to use a numerical solver to evaluate the log-likelihood values. In both, the frequentist and Bayesian paradigms, classical methods coupled with ODEs solvers were used to solve the problem. These are:

1. Non-linear least squares (NLS).
2. Markov chain monte carlo (MCMC).

NLS is a frequentist method in which the estimate of the parameter is obtained by maximizing (1.4); this is done numerically. The optimization algorithm, which begins from some initial value, updates the parameter at each step. The log-likelihood $l(\boldsymbol{\theta}, \boldsymbol{\xi})$ needs to be evaluated at some particular parameter at every step. MCMC is a Bayesian method that provides an approximation of the posterior distribution of the unknown parameter by constructing a Markov chain. The chain is initialized to some value of the parameter and then on every step a new value $(\boldsymbol{\theta}_{\text{new}}, \boldsymbol{\xi}_{\text{new}})$ is proposed according to a *proposal distribution*. That value is added to the chain or the old value $(\boldsymbol{\theta}_{\text{old}}, \boldsymbol{\xi}_{\text{old}})$ is replicated, depending on the certain probability that depends on $l(\boldsymbol{\theta}_{\text{new}}, \boldsymbol{\xi}_{\text{new}})$ and $l(\boldsymbol{\theta}_{\text{old}}, \boldsymbol{\xi}_{\text{old}})$. Thus in both approaches at every step in the procedures the log-likelihood needs to be evaluated, which involves solving the ODEs and hence the computational burden.

**Classical methods coupled with data smoothing**

Another approach is to construct an approximation of the solution based on the data, by using a smoothing method. This approach has turned out to be fruitful. The idea is old (see e.g. Bellman and Roth (1971); Varah (1982)), but considerable time passed before it was refined. The seminal paper by Ramsay et al. (2007) solved many issues and attracted the interest of researchers. The proposed approach has many strong points and it has also been explored from a Bayesian viewpoint (Campbell, 2007).

**Purely Bayesian approach: Gaussian processes**

In Bayesian statistics parameters are considered as random variables. A novel idea proposed by Calderhead et al. (2008) involves considering state variables as random with a prior distribution; the theory of Gaussian processes makes this possible. The inference involves MCMC methodology which yields a posterior distribution over the parameter and the states. This idea has been substantially improved in recent work by Chkrebtii et al. (2013).

The most important contributions in the field are given in Table 1.2. Our treatment is presented in chapters 4 and 5.

| Authors | Type of method | Error | Do all the states need to be observed? | Theoretical properties | R package |
|---|---|---|---|---|---|
| Ramsay et al. (2007) | frequentist | not only Gaussian | No | $\sqrt{n}$-consistent asymptotically normal asymptotically efficient | CollocInfer |
| Brunel (2008) | frequentist | not only Gaussian | Yes | $\sqrt{n}$-consistent asymptotically normal | N.A. |
| Calderhead et al. (2008) | Bayesian | Gaussian | No | N.A. | N.A. |
| Liang and Wu (2008) | frequentist | not only Gaussian | Yes | strongly consistent | N.A. |
| Chen and Wu (2008b) | frequentist | not only Gaussian | No | optimal convergence rate but only for a linear model | N.A. |
| Gugushvili and Klaassen (2012) | frequentist | not only Gaussian | Yes | $\sqrt{n}$-consistent | N.A. |
| Chkrebtii et al. (2013) | Bayesian | Gaussian | No | probabilistic solution of the state is consistent in $L_1$ norm | N.A. |
| Dattner and Klaassen (2013) | frequentist | not only Gaussian | Yes | $\sqrt{n}$-consistent | N.A. |
| Hall and Ma (2014) | frequentist | not only Gaussian | Yes | $\sqrt{n}$-consistent asymptotically normal | N.A. |

Table 1.2 List of general methods for solving the inverse problem of ODEs starting from the seminal paper Ramsay et al. (2007).

### 1.3.2 References

Bellman and Roth (1971), Hemker (1972), Bard (1974), Varah (1982), Voit and Savageau (1982), Bock (1983), Biegler et al. (1986), Vajda et al. (1986), Bates and Watts (1988), Wild and Seber (1989), Tjoa and Biegler (1991), Ogunnaike and Ray (1994), Gelman et al. (1996), Stortelder (1996), Ramsay (1996), Jost and Ellner (2000), Pascual and Ellner (2000), Ellner et al. (2002), Putter et al. (2002), Li et al. (2002), Madár et al. (2003), Li et al. (2005), Ramsay and B.W. (2006), Barenco et al. (2006), Lawrence et al. (2006), Lalam and Klaassen (2006), Huang et al. (2006), Huang and Wu (2006), Poyton et al. (2006), Ramsay et al. (2007), Cao and Ramsay (2007), Hooker and Biegler (2007), Campbell (2007), Khanin et al. (2007), Rogers et al. (2007), Donnet and Samson (2007), Hooker (2007), Brunel (2008), Calderhead et al. (2008), Liang and Wu (2008), Chen and Wu (2008b), Varziri et al. (2008), Cao et al. (2008), Girolami (2008), Miao et al. (2008), Chen and Wu (2008a), Cao and Zhao (2008), Steinke and Schölkopf (2008), Hooker (2009), Äijö and Lähdesmäki (2009), Secrier et al. (2009), Qi and Zhao (2010), Lillacci and Khammash (2010), Lu et al. (2011), Lawrence et al. (2011), Gugushvili and Klaassen (2012), Chkrebtii et al. (2013), Dattner and Klaassen (2013), González et al. (2013), González et al. (2014), Hall and Ma (2014).

## 1.4 Our work and contribution

In this section we outline the contributions made by this thesis. The results presented are based on the following articles and manuscripts: Vujačić et al. (2014a), Abbruzzo et al. (2014), Vujačić et al. (2014b), González et al. (2014) and González et al. (2013).

1. **Estimating Kullback-Leibler information in Gaussian graphical models**

   **Motivation**. A special feature of the Gaussian graphical model is that the estimation of its parameters, i.e. the precision matrix, can have two goals. One is prediction accuracy, which is usually measured in terms of Kullback-Leibler distance. The other is graph identification.

   **Results**. The scaled log-likelihood is a biased KL estimator and estimating the bias yields an improved estimator of KL. We propose two criteria, both having the same structure: the sum of the log-likelihood term and an estimated bias term. The first criterion, which we call Kullback-Leibler cross-validation (KLCV), is an approximation of leave-one-out cross-validation. The second cri-

terion is based on the generalized information criterion (GIC), which is a generalization of the AIC for a wider class of models. This is the content of Chapter 3, which is based on Vujačić et al. (2014a) and Abbruzzo et al. (2014).

2. **Estimating graph structure in Gaussian graphical models**
   **Motivation**. In gene regulatory networks we are usually interested in relationships between different elements. It is of interest to infer these relationships as represented by a graph. In a Gaussian graphical model framework this means that we are more interested in the graph induced by the precision matrix than in the precision matrix itself.
   **Results**. The only existing closed-form model selection consistent criteria are BIC and EBIC. They use degrees of freedom that are unstable when the sample size is small. We use the derived criteria KLCV and GIC to define alternative degrees of freedom for Gaussian graphical models. These alternative definitions can be used with BIC or EBIC when the sample size is small. We do not advocate their usage for larger sample sizes. We discuss this in Chapter 3.
3. **Estimating parameters in general ordinary differential equations**
   **Motivation**. In many sciences, dynamic processes are modelled by ordinary differential equations. These equations contain parameters that need to be estimated based on the data.
   **Results**. A widely used approach to estimating parameters in ODEs involves replacing the solution of the ODEs by its estimate obtained from the data. In Chapter 5 we use this idea but we formulate it in a framework of reproducing kernel Hilbert spaces. We consider smoothing the state as a problem of estimating a regression function that is also close to the solution of the ODEs. To make the idea possible we discretize the problem. The proposed estimator avoids the usage of numerical solvers of ODEs. The material in this chapter is based on González et al. (2014) and González et al. (2013).
4. **Estimating parameters in autonomous ordinary differential equations linear in the parameters**
   **Motivation**. In many applications the system of ordinary differential equations, which is a model of some dynamic process, is autonomous and linear in the parameters.
   **Results**. The special structure of the autonomous systems that are linear in the parameters allows one to obtain explicit estimators of the parameters and initial values. This idea has been explored in the case of repeated measurements.

For time course data we introduce a window estimator that yields $\sqrt{n}$-consistent estimators of the parameters. The explicit form of the estimators are also available in this case. Because of this, no optimization is needed and estimation is computationally fast. Due to its computational efficiency, the estimator can be combined with more efficient procedures. This is the topic of Chapter 4, which is based on Vujačić et al. (2014b).

5. **Inferring latent gene regulatory network kinetics**
   **Motivation**. Transcription factors (TFs) are proteins that activate or repress genes. Regulatory networks consist of genes and TFs and their dynamics can be modelled by ODEs. It is of interest to infer the activity profile of TFs that are unobserved, and to estimate the kinetic parameters of the ODEs using level expression measurements of the genes regulated by the TFs.
   **Results**. We model the regulatory network in *Escherichia coli*. By using the theory developed in Chapter 5, we reconstruct the unobserved LexA transcription factor and estimate the kinetic parameters of the ODEs. We also use an EM algorithm to improve the precision of the derivative approximation. This is the topic of Chapter 6, which is based on González et al. (2013).

## 1.5 Outline of the thesis

The remainder of this thesis is divided into five chapters. Chapter 2 includes background material for Chapter 3; more specifically, in Chapter 2 we give an overview of Gaussian graphical models and penalized maximum likelihood estimation as well as model selection for these models. In Chapter 3 we describe two new estimates of Kullback-Leibler information for Gaussian graphical models and we show how they can also be used for graph estimation. Chapter 4 deals with the estimation of parameters of autonomous differential equations linear in the parameters. In Chapter 5 we describe a method for estimating parameters of general differential equations which uses the theory of reproducing kernel Hilbert (RKHS) spaces. In Chapter 6 we apply the methodology developed in Chapter 5 to infer gene regulatory kinetics in the case of the SOS repair system in Escherichia coli.

# Chapter 2

# Model selection in Gaussian graphical models

A graphical model is a statistical model that uses a graph to represent conditional dependencies between random variables. Having a graphical representation of the dependencies enables one to have better understanding of the relations between random variables. To undertake a formal definition of a graphical model we first introduce a notion of conditional independence. Our exposition is based mainly on Lauritzen (1996). Other useful references are Whittaker (2009), Rue and Held (2005), Edwards (2000) and Fried and Vogel (2010). Hereafter we focus on continuous random vectors, although the theory is more general (Lauritzen, 1996). Therefore, throughout this chapter we consider the Lebesgue measure on the product space.

**Definition 2.1.** *Let $X, Y, Z$ be real random variables with a joint distribution $P$ that has a continuous density $f$ with respect to the product measure. We say that the random variable $X$ is conditionally independent of $Y$ given $Z$ under $P$, and write $X \perp\!\!\!\perp Y|Z[P]$, if*

$$f_{XY|Z}(x, y|z) = f_{X|Z}(x|z) f_{Y|Z}(y|z),$$

*for all $z$ with $f_Z(z) > 0$. If $Z$ is trivial we say that $X$ is independent of $Y$, and write $X \perp\!\!\!\perp Y[P]$.*

The simpler notation $X \perp\!\!\!\perp Y|Z$ and $X \perp\!\!\!\perp Y$ is usually used. Consider a random vector $\boldsymbol{Y} = (Y_1, \ldots, Y_p)^\top : \Omega \to \mathbb{R}^p$, where $\Omega$ is the sample space. We are interested in relations of the type

$$\boldsymbol{Y}_A \perp\!\!\!\perp \boldsymbol{Y}_B | \boldsymbol{Y}_C,$$

where $\boldsymbol{Y}_A$ stands for $(Y_i)_{i\in A}$ and $A, B, C$ are subsets of the set $V = \{1, 2, ..., p\}$. The aim is to have a graph that describes the probability distribution of the vector $\boldsymbol{Y}$. In this thesis, a graph is a pair $\mathcal{G} = (V, E)$, where $V = \{1, 2, ..., p\}$ is a finite set whose elements are called *nodes* or *vertices* and $E$ is a subset of unordered pairs of distinct values from $V$, whose elements are called *edges* or *lines*. Thus our graphs are *finite* – have a finite set of nodes, *undirected* – the edges are undirected and *simple* – there are no multiple edges and no edges that connect a node to itself (loops).

In order to bring together random vector $\boldsymbol{Y}$ and graph $\mathcal{G}$ we assign to every random variable $Y_i$ a node $i \in V$ and to any unordered pair $\{Y_i, Y_j\}$ of random variables an edge $\{i, j\} \in E$. In this context, instead of $\boldsymbol{Y}_A \perp\!\!\!\perp \boldsymbol{Y}_B | \boldsymbol{Y}_C$ we write

$$A \perp\!\!\!\perp B | C.$$

## 2.1 Undirected graphical models

From the definition of conditional independence it follows that the condition $A \perp\!\!\!\perp B | C$, can only be satisfied when the sets $A, B, C$ are disjoint. Depending on the nature of the sets $A, B, C$ we have the following so-called *Markov properties*. A probability measure $P$ on $\mathbb{R}^p$ is said to obey:

(P) the *pairwise Markov property*, relative to $\mathcal{G}$, if for any pair $(i, j)$ of non-adjacent nodes

$$i \perp\!\!\!\perp j | V \setminus \{i, j\}\,.$$

(L) the *local Markov property*, relative to $\mathcal{G}$, if for node $i \in V$

$$i \perp\!\!\!\perp V \setminus \mathrm{cl}(i) | \mathrm{ne}(i),$$

where $\mathrm{ne}(i) = \{j \in V : \{i, j\} \in E\}$ is the *set of neighbours* of a node $i$ and $\mathrm{cl}(i) = \mathrm{ne}(i) \cup \{i\}$ is the *closure* of the set $\{i\}$.

(G) the *global Markov property*, relative to $\mathcal{G}$ if, for any triple $(A, B, C)$ of disjoint subsets of $V$ such that $C$ separates $A$ from $B$ in $\mathcal{G}$

$$A \perp\!\!\!\perp B | C.$$

The subset $C$ is said to *separate* $A$ from $B$ if all paths in $\mathcal{G}$ from any $a \in A$ to any $b \in B$ intersect $C$. A path in $\mathcal{G}$ is a sequence of distinct nodes such that two

consecutive nodes form an edge in $\mathcal{G}$.

The properties implicitly describe the distribution $P$; they do not define the form of the density of $P$, in case it exists. The next property does exactly that.

(F) A probability measure $P$ on $\mathbb{R}^p$ is said to *factorize* according to $\mathcal{G}$, if for all cliques $c \subset V$ there exist non-negative functions $\psi_c$ that depend on $\boldsymbol{y} = (y_1, ..., y_p)$ through $\boldsymbol{y}_c = (y_i)_{i \in c}$ only, such that the density $f$ of $P$ has the form

$$f(\boldsymbol{y}) = \prod_{c \in C} \psi_c(\boldsymbol{y}_c),$$

where $C$ is a set of cliques. A *clique* is a maximal subset of nodes (with respect to $\subseteq$) that has the property that each two nodes in the subset are joined by a line.

For any undirected graph $\mathcal{G}$ and any probability distribution on $\mathbb{R}^p$ it holds that $(F) \Rightarrow (G) \Rightarrow (L) \Rightarrow (P)$. More can be said for the distributions that have continuous positive density; that is the content of the following theorem.

**Theorem 2.1** (**Hammersley and Clifford**). *A probability distribution with positive and continuous density function with respect to the product measure satisfies the pairwise Markov property with respect to an undirected graph $\mathcal{G}$ if and only if it factorizes according to $\mathcal{G}$.*

The theorem implies that in the case of a probability distribution with positive and continuous density with respect to a product measure it holds that:

$$(F) \Leftrightarrow (G) \Leftrightarrow (L) \Leftrightarrow (P).$$

Finally, we give the definition of an undirected graphical model (Wainwright and Jordan, 2008).

**Definition 2.2.** *An undirected graphical model - also known as a Markov random field (MRF) – associated with graph $\mathcal{G}$ is a family of probability distributions that factorizes according to $\mathcal{G}$.*

The definition assumes the strongest $F$ property, although some authors assume the weakest, pairwise Markov property (Whittaker, 2009).

**Definition 2.3.** *An undirected graphical model associated with graph $\mathcal{G}$ is a family of probability distributions that satisfies the pairwise conditional independence restrictions inherent in $\mathcal{G}$.*

In the case of probability distributions that have positive and continuous density, the case considered in this thesis, theorem 2.1 implies that both definitions are equivalent.

## 2.2 Gaussian graphical models

If in the definition of the graphical model we restrict the family of distributions to be Gaussian we obtain a *Gaussian graphical model* (GGM). In the case of a GGM the conditional dependencies can be read off from the inverse covariance matrix which is called the *precision matrix* or *concentration matrix*.

The density of the normal distribution with mean $\boldsymbol{\mu}$ and precision matrix $\boldsymbol{\Theta} = (\theta_{ij})$ has the form

$$f(\boldsymbol{y}) = (2\pi)^{-p/2}|\boldsymbol{\Theta}|^{1/2} \exp\left\{-(\boldsymbol{y}-\boldsymbol{\mu})^\top \boldsymbol{\Theta}(\boldsymbol{y}-\boldsymbol{\mu})/2\right\}.$$

The following result is of fundamental importance.

**Proposition 2.1.** *For a Gaussian graphical model it holds*

$$i \perp\!\!\!\perp j | V \setminus \{i,j\} \iff \theta_{ij} = 0.$$

The consequence of the proposition is that the precision matrix contains all the information about conditional independence relations between the variables. The edge between two nodes in the conditional independence graph is present if and only if the element in the precision matrix determined by the two nodes is not equal to zero. For example, the following precision matrix and graph correspond to each other, where $*$ represents a non-zero element.

$$\boldsymbol{\Theta} = \begin{pmatrix} * & * & 0 & * \\ * & * & * & 0 \\ 0 & * & * & * \\ * & 0 & * & * \end{pmatrix} \qquad \begin{array}{ccc} 2 & - & 3 \\ | & & | \\ 1 & - & 4 \end{array}$$

Now we formulate some basic results in regard to the GGM.

Assume that the data $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$ are i.i.d. sample from $\mathcal{N}_p(\boldsymbol{0}, \boldsymbol{\Theta}^{-1})$; for simplicity we assume that the mean is zero. Using the notation $\mathbf{S}_k = \boldsymbol{y}_k \boldsymbol{y}_k^\top$ for the empirical covariance matrix of a single observation, we have that the empirical covariance matrix

is given as $\mathbf{S} = \sum_{k=1}^{n} \mathbf{S}_k/n$. The log-likelihood of one observation $\boldsymbol{y}_k$ is up to an additive constant

$$l_k(\boldsymbol{\Theta}) = \frac{1}{2}\left\{\log|\boldsymbol{\Theta}| - \mathrm{tr}(\boldsymbol{\Theta}\mathbf{S}_k)\right\},$$

and the log-likelihood of the data is

$$l(\boldsymbol{\Theta}) = \sum_{k=1}^{n} l_k(\boldsymbol{\Theta}) = \frac{n}{2}\{\log|\boldsymbol{\Theta}| - \mathrm{tr}(\boldsymbol{\Theta}\mathbf{S})\}. \tag{2.1}$$

For the result that follows we introduce some notation. The commutation matrix $\mathbf{K}_p$ is defined as a matrix that has the property $\mathbf{K}_p\mathrm{vec}\mathbf{A} = \mathrm{vec}\mathbf{A}^\top$ for any $p \times p$ matrix $\mathbf{A}$. Here, vec is the vectorization operator which transforms a matrix into a column vector obtained by stacking the columns of the matrix on top of one another. Define $\mathbf{M}_p = (\mathbf{I}_{p^2} + \mathbf{K}_p)/2$, where $\mathbf{I}_{p^2}$ is identity matrix of order $p^2$.

**Proposition 2.2.** *If $n > p$ the maximum likelihood estimator (MLE) almost surely exists and is given by $\widehat{\boldsymbol{\Theta}} = \mathbf{S}^{-1}$. Furthermore, $\widehat{\boldsymbol{\Theta}}$ is a strongly consistent estimator of the true precision matrix $\boldsymbol{\Theta}_0$ and*

$$\sqrt{n}(\widehat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}_0) \to \mathcal{N}_{p^2}\left(0, 2\mathbf{M}_p(\boldsymbol{\Theta}_0 \otimes \boldsymbol{\Theta}_0)\right), \qquad n \to \infty.$$

For details see Fried and Vogel (2010).

## 2.3 Kullback-Leibler information

In this section we review the Kullback-Leibler information which is used as a criterion for evaluating statistical models. Assume that the data $\mathcal{D} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\}$ are i.i.d. sample generated from some $p$-dimensional distribution $q$ that we refer to as the *true distribution*. Let $\Theta \subset \mathbb{R}^p$ and consider a parametric family of distributions $\{p(\boldsymbol{y}|\boldsymbol{\theta}); \boldsymbol{\theta} \in \Theta\}$ that we use to approximate the true distribution.

The goodness of the model $p(\boldsymbol{y})$ can be assessed in terms of its closeness to the true distribution $q(\boldsymbol{y})$. Akaike (1973) proposed to measure this closeness by using *Kullback-Leibler information* [or *Kullback-Leibler divergence*, Kullback and Leibler (1951), hereinafter abbreviated as "KL"]:

$$\mathrm{KL}(q; p) = \mathrm{E}_q\left\{\log\frac{q(\boldsymbol{Y})}{p(\boldsymbol{Y})}\right\} = \int_{\mathbb{R}^p} \log\left\{\frac{q(\boldsymbol{y})}{p(\boldsymbol{y})}\right\} q(\boldsymbol{y})d\boldsymbol{y}.$$

Here, $\boldsymbol{Y}$ stands for the random variable distributed according to $p(\boldsymbol{y})$. The basic properties of KL are given in the following proposition (Konishi and Kitagawa, 2008).

**Proposition 2.3** (Properties of KL.)**.** *The KL has the following properties:*

1. $\mathrm{KL}(q;p) \geq 0$,
2. $\mathrm{KL}(q;p) = 0 \Leftrightarrow q(\boldsymbol{y}) = p(\boldsymbol{y})$.

KL is not a metric on the space of probability distributions since it is not symmetric and does not satisfy the triangle inequality. KL is, up to a constant, equal to the minus expected log-likelihood since

$$\mathrm{KL}(q;p) = \mathrm{E}_q\left\{\log\frac{q(\boldsymbol{Y})}{p(\boldsymbol{Y})}\right\} = \mathrm{E}_q\{\log q(\boldsymbol{Y})\} - \mathrm{E}_q\{\log p(\boldsymbol{Y})\} = C - \mathrm{E}_q\{\log p(\boldsymbol{Y})\},$$

where $C = \mathrm{E}_q\{\log q(\boldsymbol{Y})\}$ does not depend on $p$. Assume that $p$ is chosen from a parametric family of distributions, i.e. $p(\cdot) = p(\cdot|\widehat{\boldsymbol{\theta}})$, where $\widehat{\boldsymbol{\theta}}$ is an estimator of $\boldsymbol{\theta}$. Denote by $l(\boldsymbol{\theta}|\mathcal{D}_S)$ the log-likelihood function based on the data set $\mathcal{D}_S \subset \mathcal{D}$ and let $l_k(\boldsymbol{\theta}) = l(\boldsymbol{\theta}|\boldsymbol{y}_k)$ and $l(\boldsymbol{\theta}) = l(\boldsymbol{\theta}|\mathcal{D})$ denote the log-likelihood based on the $k$th observation and the entire data set, respectively. Estimating $\mathrm{E}_q\{\log p(\boldsymbol{y}|\widehat{\boldsymbol{\theta}})\}$ by replacing the density $q(\boldsymbol{y})$ by its empirical counterpart we obtain that

$$\widehat{\mathrm{KL}}(q(\cdot);p(\cdot|\widehat{\boldsymbol{\theta}})) = -\frac{1}{n}\sum_{k=1}^{n}\log p(\boldsymbol{y}_k|\widehat{\boldsymbol{\theta}}) = -\frac{1}{n}\sum_{k=1}^{n} l_k(\widehat{\boldsymbol{\theta}}) = -\frac{1}{n}l(\widehat{\boldsymbol{\theta}}).$$

Thus the scaled log-likelihood is an estimator of KL. However, it is a biased estimator since the data have been used twice – once to obtain the estimator $\widehat{\boldsymbol{\theta}}$ and once to obtain the empirical density of the true distribution. We list some of the estimators of KL that reduce this bias.

**Akaike's Information Criterion (AIC)** *Akaike's Information Criterion*, introduced by Akaike (1973), is an estimator of KL in the case of the models estimated by maximum likelihood. It reduces the bias by adding the penalty to the likelihood, where the penalty is given in terms of the number of the parameters of the model. AIC has the form:

$$\mathrm{AIC} = -2l(\widehat{\boldsymbol{\theta}}) + 2\mathrm{df},$$

where $\mathrm{df} = \dim(\Theta)$. Similarly, in case of penalized maximum likelihood estimators we can define the AIC selector (Zhang et al., 2010), which is a special case of Nishii's *generalized information criterion* (GIC) (Nishii, 1984). The AIC selector has the form:

$$\text{AIC} = -2l(\widehat{\boldsymbol{\theta}}_\lambda) + 2\text{df}(\lambda), \tag{2.2}$$

where $\widehat{\boldsymbol{\theta}}_\lambda$ is the penalized maximum likelihood estimator and $\text{df}(\lambda)$ is the degrees of freedom of the corresponding model. We refer to these criteria simply as AIC.

**Generalized Information Criterion (GIC)** *Generalized Information Criterion*, introduced by (Konishi and Kitagawa, 1996), is an estimator of KL which is applicable to a wide class of models and not only for models estimated by maximum likelihood. This is different from the GIC proposed by (Nishii, 1984). Here we present the GIC for $M$ estimators. An *$M$-estimator* is defined as a solution of the system of equations

$$\sum_{k=1}^{n} \boldsymbol{\psi}(\boldsymbol{y}_k, \boldsymbol{\theta}) = \mathbf{0}_d,$$

where $\boldsymbol{\psi}$ is a column vector of dimension $d$ and $\mathbf{0}_d$ is the zero vector of the same dimension. The GIC for an $M$-estimator (Konishi and Kitagawa, 2008) is given by:

$$\text{GIC} = -2l(\widehat{\boldsymbol{\theta}}) + 2\text{tr}(\mathbf{R}^{-1}\mathbf{Q}), \tag{2.3}$$

where $\mathbf{R}$ and $\mathbf{Q}$ are square matrices of order $p$ given by

$$\mathbf{R} = -\frac{1}{n}\sum_{k=1}^{n}\{\mathsf{D}\boldsymbol{\psi}(\boldsymbol{y}_k, \widehat{\boldsymbol{\theta}})\}^\top,$$

$$\mathbf{Q} = \frac{1}{n}\sum_{k=1}^{n}\boldsymbol{\psi}(\boldsymbol{y}_k, \widehat{\boldsymbol{\theta}})\mathsf{D}l_k(\widehat{\boldsymbol{\theta}}).$$

Here, $\mathsf{D}\boldsymbol{\psi}(\boldsymbol{y}_k, \widehat{\boldsymbol{\theta}})$ and $\mathsf{D}l_k(\widehat{\boldsymbol{\theta}})$ are Jacobian matrices of corresponding functions at $\widehat{\boldsymbol{\theta}}$. We use the definition of the derivative given in Magnus and Neudecker (2007) (see appendix 3.B on matrix calculus). The maximum likelihood estimator is an $M$-estimator, corresponding to $\boldsymbol{\psi}(\boldsymbol{y}_k, \boldsymbol{\theta}) = \text{vec}\mathsf{D}l_k(\boldsymbol{\theta})$.

**Cross-validation (CV)** *Cross-validation* (Stone, 1974) is an estimator of KL that involves reusing the data. We split the data $\mathcal{D}$ into $K$, roughly equal-sized parts, that we denote by $\mathcal{D}_1, \ldots, \mathcal{D}_K$. For the $k$th part of the data $\mathcal{D}_k$, we estimate the parameter $\boldsymbol{\theta}$ by using the data from the other $K-1$ parts. Denote this estimator by $\widehat{\boldsymbol{\theta}}^{-(k)}$. Then we calculate the minus log-likelihood based on $\mathcal{D}_k$ at the estimate $\widehat{\boldsymbol{\theta}}^{-(k)}$; this is an estimator of KL. This procedure is repeated for $k = 1, 2, \ldots, K$ and then the

estimators are averaged. In other words, the cross-validation estimator of KL is

$$\text{CV} = -\frac{1}{K}\sum_{k=1}^{K} l(\widehat{\boldsymbol{\theta}}^{-(k)}|\mathcal{D}_k). \tag{2.4}$$

Usual choices of $K$ are 5 or 10. When $K = N$ we obtain *leave-one-out cross-validation* (LOOCV)

$$\text{LOOCV} = -\frac{1}{n}\sum_{k=1}^{n} l(\widehat{\boldsymbol{\theta}}^{-(k)}|\boldsymbol{y}_k) = -\frac{1}{n}\sum_{k=1}^{n} l_k(\widehat{\boldsymbol{\theta}}^{-(k)}). \tag{2.5}$$

We finish the section by presenting the KL between two normal distributions and writing the explicit form of the bias.

**KL for normal models.** Let $q(\boldsymbol{y}) = \mathcal{N}_d(\boldsymbol{y}; 0, \boldsymbol{\Theta}_q^{-1})$ and $p(\boldsymbol{y}) = \mathcal{N}_d(\boldsymbol{y}; 0, \boldsymbol{\Theta}_p^{-1})$ be the multivariate normal densities. Then (Penny, 2001):

$$\text{KL}(q; p) = \frac{1}{2}\{-\log|\boldsymbol{\Theta}_q^{-1}\boldsymbol{\Theta}_p| + \text{tr}(\boldsymbol{\Theta}_q^{-1}\boldsymbol{\Theta}_p) - d\}. \tag{2.6}$$

Having in mind expression for the log-likelihood in Gaussian models (see 2.1) and that it is a biased estimator of KL we can write

$$\text{KL}(q; p) = -\frac{1}{n} l(\boldsymbol{\Theta}_p) + \text{bias}, \tag{2.7}$$

where $\text{bias} = \text{tr}\{\boldsymbol{\Theta}_p(\boldsymbol{\Theta}_q^{-1} - \mathbf{S})\}/2$.

## 2.4 Penalized likelihood estimation in Gaussian Graphical Models

### 2.4.1 Estimation

Suppose we have $n$ i.i.d. multivariate observations $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$ from distribution $\mathcal{N}_p(0, \boldsymbol{\Theta}^{-1})$. When $n > p$ the precision matrix $\boldsymbol{\Theta}$ can be estimated by maximizing the scaled log-likelihood function (see 2.1)

$$\frac{2}{n} l(\boldsymbol{\Theta}) = \log|\boldsymbol{\Theta}| - \text{tr}(\boldsymbol{\Theta}\mathbf{S}),$$

over positive definitive matrices $\boldsymbol{\Theta}$. The global maximizer, called the maximum likelihood estimator, almost surely exists and is given by $\widehat{\boldsymbol{\Theta}} = \mathbf{S}^{-1}$ (Proposition 2.2). When $n \leq p$ a maximum likelihood estimator does not exist. Moreover, in the case when

$n > p$ and the true precision matrix is known to be sparse, the MLE has a non-desirable property: with probability one all elements of the precision matrix are nonzero. A sparse estimator can be obtained by maximizing the penalized log-likelihood:

$$\widehat{\mathbf{\Theta}}_\lambda = \text{argmax}_{\mathbf{\Theta}} \log|\mathbf{\Theta}| - \text{tr}(\mathbf{\Theta}\mathbf{S}) - \sum_{i=1}^{p}\sum_{j=1}^{p} p_{\lambda_{ij}}(|\theta_{ij}|), \tag{2.8}$$

where the maximization is over positive definitive matrices $\mathbf{\Theta}$. Here, $p_{\lambda_{ij}}$ is a sparsity inducing penalty function, $\theta_{ij}$ is the $(i,j)$th entry of matrix $\mathbf{\Theta}$ and $\lambda_{ij} > 0$ is the corresponding regularization parameter. We refer to $\boldsymbol{\lambda} = (\lambda_{11}, \ldots, \lambda_{pp})$ as a *regularization parameter* and typically $\lambda_{11} = \ldots = \lambda_{pp} = \lambda$.

The LASSO penalty uses the $L_1$ penalty function $p_\lambda(|\theta|) = \lambda|\theta|$. Friedman et al. (2008) propose the *graphical lasso* algorithm for the optimization in (2.8) in the case of the LASSO penalty. The algorithm uses a coordinate descent procedure and is extremely fast. However, it is known that the LASSO penalty produces substantial biased estimates in a regression setting (Fan and Li, 2001). To address this issue Lam and Fan (2009) have studied the theoretical properties of sparse precision matrix estimation via a general penalty function satisfying the properties in Fan and Li (2001). This general penalty function includes LASSO and also SCAD and adaptive LASSO penalties, originally introduced in a linear regression setting (Fan and Li, 2001; Zou, 2006). They also show that the LASSO penalty produces bias in the case of sparse precision matrix estimation. The SCAD penalty has a derivative of the form

$$p'_{\lambda,a}(|\theta|) = \lambda I(|\theta| \leq \lambda) + \frac{(a\lambda - |\theta|)_+}{a-1} I(|\theta| > \lambda),$$

where $a$ is a constant usually set to 3.7 (Fan and Li, 2001). Another penalty, called the adaptive LASSO, which is proposed in Zou (2006), uses the $L_1$ penalty function $p_\lambda(|\theta|) = \lambda|\theta|$ and

$$\lambda_{ij} = \lambda/|\widetilde{\theta}_{ij}|^\gamma,$$

where $\gamma > 0$ is a constant and $\widetilde{\theta}_{ij}$ is any consistent estimator of $\theta_{ij}$. The constant $\gamma$ is usually chosen to be 0.5. Implementing the optimization with the SCAD and adaptive LASSO penalties can be done efficiently by using the graphical lasso algorithm (Fan et al., 2009).

These penalties are important because under certain conditions the estimated precision matrix tends to the true one when sample size tends to infinity. Also, the estimator has the *sparsistency* property, which means that when sample size tends to

infinity all the parameters that are zero are actually estimated as zero with probability tending to one (Lam and Fan, 2009). Also Fan et al. (2009) show that using the SCAD penalty in a GGM setting produces estimators that are not just sparsistent but also $\sqrt{n}$-consistent, asymptotically normal, and have the oracle property. They also prove that oracle property holds for the adaptive LASSO penalty. *Oracle property* of the estimator means that the estimator performs as if the correct submodel is known in advance. More specifically, when there are zeros in the true precision matrix, they are estimated as zero with probability tending to one, and the nonzero components are estimated as if the correct submodel is known.

We refer to a Gaussian graphical model estimated by penalized maximum likelihood as a *penalized Gaussian graphical model*.

### 2.4.2 Model selection

Model selection in penalized graphical models is essentially a matter of choosing the regularization parameter. It has been shown that certain asymptotic rates of $\lambda$ result in consistent or sparsistent estimators (Ravikumar et al., 2011; Lam and Fan, 2009). However, for finite $n$ and $p$ the choices are less clear. In this section we review existing methods for selection of $\lambda$.

In what follows, we use the notion of *degrees of freedom* in Gaussian graphical models, defined as (Yuan and Lin, 2007):

$$\mathrm{df}(\lambda) = \sum_{1 \leq i < j \leq p} I(\widehat{\theta}_{ij,\lambda} \neq 0),$$

where $\widehat{\theta}_{ij,\lambda}$ is $(i,j)$th entry of the estimated precision matrix $\widehat{\mathbf{\Theta}}_\lambda$ and $I$ is the indicator function.

1. **Bayesian Information Criterion (BIC)** (Yuan and Lin, 2007; Schmidt, 2010; Menéndez et al., 2010; Lian, 2011; Gao et al., 2012) has the form

   $$\mathrm{BIC}(\lambda) = -2l(\widehat{\mathbf{\Theta}}_\lambda) + \log n \mathrm{df}(\lambda).$$

   Roughly speaking, the criterion provides an approximation of the natural logarithm of the posterior probability of the model defined by $\widehat{\mathbf{\Theta}}_\lambda$ scaled with -2. Approximation is based on the assumption that when $n$ tends to infinity $p$ is fixed.
2. **Extended Bayesian Information Criterion (EBIC)** (Foygel and Drton,

2010; Gao et al., 2012) is an extension of BIC and is introduced to deal with cases when also $p$ tends to infinity together with $n$. It has the form

$$\text{EBIC}(\lambda) = -2l(\widehat{\boldsymbol{\Theta}}_\lambda) + (\log n + 4\gamma \log p)\text{df}(\lambda),$$

where $\gamma \in [0, 1]$ is the parameter that penalizes the number of models, which increases when $p$ increases. In case of $\gamma = 0$ the classical BIC is obtained. Typical values for $\gamma$ are $1/2$ and 1.

3. **Aikaike's Information Criterion (AIC)** (Menéndez et al., 2010; Liu et al., 2010; Lian, 2011) has the form c.f. (2.2)

$$\text{AIC}(\lambda) = -2l(\widehat{\boldsymbol{\Theta}}_\lambda) + 2\text{df}(\lambda).$$

4. **Cross-validation (CV)** (Rothman et al., 2008; Fan et al., 2009; Schmidt, 2010; Ravikumar et al., 2011; Fitch, 2012) has the form c.f. (2.4)

$$\text{CV}(\lambda) = -\frac{1}{n}\sum_{k=1}^{K} n_k \left[\log|\widehat{\boldsymbol{\Theta}}_\lambda^{(-k)}| - \text{tr}\left\{\mathbf{S}^{(k)}\widehat{\boldsymbol{\Theta}}_\lambda^{(-k)}\right\}\right],$$

where $n_k$ is the sample size of the $k$th partition, $\widehat{\boldsymbol{\Theta}}_\lambda^{(-k)}$ is the estimator of the precision matrix, with $\lambda$ as regularization parameter, based on the data with $k$th partition excluded and $\mathbf{S}^{(k)}$ is the empirical covariance matrix based on the $k$th partition of the data. The formula presented here differs from the one given, for example, in (Fan et al., 2009; Lian, 2011) in that we have the term $-1/n$ in front of the sum. This term is usually omitted for the sake of simplicity.

5. **Generalized Approximate Cross Validation (GACV)** (Lian, 2011) is introduced as an approximation of leave-one-out cross validation (LOOCV) and has the form

$$\text{GACV}(\lambda) = -2l(\widehat{\boldsymbol{\Theta}}_\lambda) + 2\sum_{k=1}^{n} \text{vec}(\widehat{\boldsymbol{\Theta}}_\lambda^{-1} - \mathbf{S}_k)^\top \text{vec}\{\widehat{\boldsymbol{\Theta}}_\lambda(\mathbf{S}^{(-k)} - \mathbf{S})\widehat{\boldsymbol{\Theta}}_\lambda\},$$

where $\mathbf{S}^{(-k)}$ is the empirical covariance matrix based on the data with $k$th observation excluded. The formula presented here differs from the one given in Lian (2011) in that we have scaled it with $-2$. We introduce this scaling to make the connection with KL.

For any criterion mentioned above, denoted by CRITERION, the best regular-

ization parameter is chosen as

$$\lambda^* = \text{argmin}_\lambda \text{CRITERION}(\lambda).$$

6. **Stability Approach to Regularization Selection (StARS)** (Liu et al., 2010) is based on the idea of subsampling. First, define $\Lambda = 1/\lambda$ so that small $\Lambda$ corresponds to a more sparse graph. Let $G_n = \{\Lambda_1, \ldots, \Lambda_K\}$ be a grid of regularization parameters. Let $b = b(n)$ be such that $1 < b(n) < n$. We draw $N$ random subsamples $\mathcal{D}_1, \ldots, \mathcal{D}_N$ from $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$ each of size $b$. The estimation of a graph for each $\Lambda \in G_n$ yields $N$ estimated edge matrices $\widehat{E}_1^b(\Lambda), \ldots, \widehat{E}_N^b(\Lambda)$. Let $\psi^\Lambda(\cdot)$ denote the graph estimation algorithm with the regularization parameter $\Lambda$ and for any subsample $\mathcal{D}_j$ let $\psi_{st}^\Lambda(\cdot)$ be equal to one if the estimation algorithm puts an edge $\{s, t\}$, and otherwise let it be equal to zero. Define the parameters $\theta_{st}^b(\Lambda) = P(\psi_{st}^\Lambda(\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_b) = 1)$ and $\xi_{st}^b(\Lambda) = 2\theta_{st}^b(\Lambda)\{1 - \theta_{st}^b(\Lambda)\}$ and its estimators $\widehat{\theta}_{st}^b(\Lambda) = \frac{1}{N}\sum_{j=1}^N \psi_{st}^\Lambda(\mathcal{D}_j)$ and $\widehat{\xi}_{st}^b(\Lambda) = 2\widehat{\theta}_{st}^b(\Lambda)\{1 - \widehat{\theta}_{st}^b(\Lambda)\}$, respectively. Then $\xi_{st}^b(\Lambda)$ is the variance of the Bernoulli indicator of the edge $\{s, t\}$ as well as the fraction of times each pair of graphs disagree on the presence of the edge. For $\Lambda \in G_n$, the quantity $\xi_{st}^b(\Lambda)$ measures instability of the edge across subsamples, with $0 \leq \xi_{st}^b(\Lambda) \leq 1/2$. Total instability is defined by averaging over all edges $\widehat{D}_b(\Lambda) = \Sigma_{s<t}\widehat{\xi}_{st}^b / \binom{p}{2}$. Let $\overline{D}_b(\Lambda) = \sup_{0 \leq t \leq \Lambda} \widehat{D}_b(t)$, then the StARS approach chooses $\Lambda$ by defining

$$\widehat{\Lambda}_s = \sup\{\Lambda : \overline{D}_b(\Lambda) \leq \beta\}$$

for specified cut point value $\beta$, usually taken to be equal to 0.05.

### 2.4.3 Measuring the goodness of a model

The goodness of the estimated precision matrix can be evaluated in terms of: (i) graph accuracy, i.e. how close its corresponding graph is to the true graph or (ii) prediction accuracy. BIC, EBIC and StARS are designed for the former and CV, GACV and AIC for the latter.

To measure graph accuracy we use $\text{F}_1$ measure

$$\text{F}_1 = \frac{2\text{TP}}{2\text{TP} + \text{FN} + \text{FP}},$$

where *True and False Positives* (TP/FP) refer to the estimated edges that are correct/incorrect; *True and False Negatives* (TN/FN) are defined in a similar way (Baldi et al., 2000; Powers, 2011). The $F_1$ score measures the quality of a binary classifier by taking into account both true positives and negatives and takes values in the interval $[0, 1]$. The larger the value of this measure, the better the classifier. The prediction accuracy is measured by KL-information.

Depending on whether we are interested in graph identification or prediction we should use different methods for choosing the regularization parameter (see Section 3.1).

If the aim is graph estimation then the criteria BIC, EBIC and StARS are appropriate. Most of them are graph selection consistent, which means that when the sample size goes to infinity they identify the true graph. The BIC is shown to be consistent for penalized graphical models with adaptive LASSO and SCAD penalties for fixed $p$ (Lian, 2011; Gao et al., 2012). Numerical results suggest that the BIC is not consistent with the LASSO penalty (Foygel and Drton, 2010; Gao et al., 2012). When also $p$ tends to infinity EBIC is shown to be consistent for the graphical LASSO, though only for decomposable graphical models (Foygel and Drton, 2010). The disadvantage of EBIC is that it includes an additional parameter $\gamma$ that needs to be tuned. Gao et al. (2012) set $\gamma$ to one and show that in this case the EBIC is consistent with the SCAD penalty. StARS has the property of partial sparsistency which means that when the sample size goes to infinity all the true edges are included in the selected model (Liu et al., 2010).

On the other hand, Cross-validation (CV), Generalized Approximate Cross Validation (GACV) and AIC are methods for evaluating prediction accuracy. Cross-validation and AIC are both estimators of Kullback-Leibler (KL) loss (Yanagihara et al., 2006), which under some assumptions are asymptotically equivalent (Stone, 1977). Generalized Approximate Cross Validation is also an estimator of KL loss since it is derived as an approximation to leave-one-out cross-validation (Lian, 2011). The advantage of AIC and GACV is that they are less computationally expensive than CV. In the next chapter we propose two new estimators of Kullback-Leibler loss. We also show how they can be used for graph estimation.

## 2.5 Summary

In this chapter we have reviewed undirected graphical models which are widely used to model conditional independence relationships. We focused on a special class, Gaussian graphical models (GGMs), for which the pattern of zeros in the precision matrix determines the conditional independence relationships. When the number of variables is large as compared to the sample size, we have introduced the penalized likelihood estimation of GGMs. The Kullback-Leibler (KL) information was introduced as a criterion to evaluate statistical models. Various estimators of KL, such as the Akaike's Information Criterion (AIC), Cross Validation (CV) and Generalized Approximate Cross Validation (GACV) for Gaussian graphical models were presented. These criteria yield an estimator of a precision matrix which has good predictive power. For graph estimation we presented the Bayesian Information Criterion (BIC), Extended Bayesian Information Criterion (EBIC) and Stability Approach to Regularization Selection (StARS).

# Chapter 3

# Estimating the KL loss in Gaussian graphical models

In this chapter, we propose two estimators of the Kullback-Leibler loss in Gaussian graphical models. One approach uses the Generalized Information Criterion (GIC) and the other, which we call Kullback Leibler cross-validation (KLCV), is based on deriving a closed form approximation of leave-one-out cross validation. We first derive the formulae for the maximum likelihood estimator (MLE) for both GIC and KLCV. For the maximum penalized likelihood estimator (MPLE) we use a unifying framework which allows us to modify the formulae derived for the MLE so as to incorporate the assumption of sparsity. As pointed out in the previous chapter, in GGM a distinction should be made between estimating the KL and estimating the graph. Consequently, we treat the graph estimation problem separately. We explore the use of the proposed criteria in the graph estimation problem by combining it with consistent graph selection criteria such as BIC and EBIC.

The remainder of this chapter is divided into seven sections. In Section 3.1 we clarify the aim of different model selection methods. In Section 3.2 we introduce two new estimators of KL loss for which the derivation is given in Section 3.3. Section 3.4 deals with the computational aspect of the proposed methods, while Section 3.5 shows their performance on simulated data. Finally, the use of the proposed criteria for graph estimation is explored in Section 3.6. The last section contains the summary. All proofs and auxiliary results are given in the Appendix.

## 3.1 Prediction VS graph structure

Let $\boldsymbol{\Theta}$ be a precision matrix that corresponds to the true non-complete graph $\mathcal{G}$ and let $\boldsymbol{\Theta}_\epsilon$ be the matrix obtained by adding $\epsilon > 0$ to every entry of matrix $\boldsymbol{\Theta}$. The matrix $\boldsymbol{\Theta}_\epsilon$ is positive definite since it is a sum of one positive definite matrix and one positive semi-definite matrix. Indeed, $\boldsymbol{\Theta}_\epsilon = \boldsymbol{\Theta} + \boldsymbol{x}_\epsilon \boldsymbol{x}_\epsilon^\top$, where $\boldsymbol{x}_\epsilon = (\sqrt{\epsilon}, \ldots, \sqrt{\epsilon})^\top$ is a vector of dimension $p$. Hence, $\boldsymbol{\Theta}_\epsilon$ belongs to the class of precision matrices and it corresponds to a graph $\mathcal{G}_\epsilon$. The Kullback-Leibler divergence of $\mathcal{N}(\mathbf{0}, \boldsymbol{\Theta}_\epsilon^{-1})$ from $\mathcal{N}(\mathbf{0}, \boldsymbol{\Theta}^{-1})$, denoted by $\mathrm{KL}(\boldsymbol{\Theta}; \boldsymbol{\Theta}_\epsilon)$, is equal to

$$\mathrm{KL}(\boldsymbol{\Theta}; \boldsymbol{\Theta}_\epsilon) = \frac{1}{2}\{\mathrm{tr}(\boldsymbol{\Theta}^{-1}\boldsymbol{\Theta}_\epsilon) - \log|\boldsymbol{\Theta}^{-1}\boldsymbol{\Theta}_\epsilon| - p\}$$

(see 2.6). Since $\epsilon \to 0$ implies $\boldsymbol{\Theta}_\epsilon \to \boldsymbol{\Theta}$, by continuity of the log-determinant and trace it follows that

$$\lim_{\epsilon \downarrow 0} \mathrm{KL}(\boldsymbol{\Theta}; \boldsymbol{\Theta}_\epsilon) = 0.$$

However, for every $0 < \epsilon < \min_{i,j} |\theta_{ij}|$ the matrix $\boldsymbol{\Theta}_\epsilon$ is a matrix without zero entries and consequently the graph $\mathcal{G}_\epsilon$ is the full graph. Thus, even though a matrix can be close to the precision matrix of the true distribution with respect to KL loss, the corresponding graph can be completely different from the true one.

Since $K$-CV, AIC and GACV are estimators of KL, their use is appropriate for obtaining a model with good predictive power. On the other hand, the previous example indicates that they should not be used for graph identification. The following theorem confirms this claim in the case of the AIC when $p$ is fixed as $n \to \infty$. To formulate the theorem we introduce some notation and assumptions. Each precision matrix induces a set of labels corresponding to its nonzero entries. The full model $\bar{\alpha} = \{(i,j) : i,j = 1,\ldots,p;\, i < j\}$ is the set of labels of all entries in the upper diagonal of the corresponding precision matrix; it corresponds to precision matrices without nonzero elements. A *candidate model* is a set $\alpha \subset \bar{\alpha}$ that consists of labels of nonzero elements of the corresponding precision matrix. We denote by $\mathcal{A}$ the collection of all candidate models and assume that there is a unique true model $\alpha_0$ in $\mathcal{A}$. Let $\Omega_- = \{\lambda : \alpha_\lambda \not\supset \alpha_0\}$, $\Omega_0 = \{\lambda : \alpha_\lambda = \alpha_0\}$ and $\Omega_+ = \{\lambda : \alpha_\lambda \supset \alpha_0 \text{ and } \alpha_\lambda \neq \alpha_0\}$ denote the sets of regularization parameters that induce, respectively, underfitted, true, and overfitted model. We assume technical conditions as in Zhang et al. (2010), from which the proof is adapted. These conditions are satisfied by SCAD and $L_1$ penalties.

**Theorem 3.1.** *Assume that $p$ is fixed as $n \to \infty$ and denote the optimal tuning parameter selected by minimizing* $\mathrm{AIC}(\lambda)$ *by* $\widehat{\lambda}_{\mathrm{AIC}}$. *Then the penalized likelihood estimator defined in 2.8 cannot correctly identify all the zero elements in the true precision matrix. That is,*

$$P(\widehat{\lambda}_{\mathrm{AIC}} \in \Omega_-) \to 0 \qquad \text{and} \qquad P(\widehat{\lambda}_{\mathrm{AIC}} \in \Omega_+) \to \pi,$$

*where $\pi$ is a nonzero probability.*

The proof of the theorem is given in the Appendix 3.A. We conjecture that the theorem also holds in the case when $p \to \infty$ as $n \to \infty$. Moreover, we hypothesize that GACV and $K$-CV, where $K$ is fixed as $n \to \infty$ are not graph selection consistent.

For graph identification, BIC, EBIC and StARS are appropriate because of their graph selection consistency. Consequently, we treat these two problems separately. We devote the next section to two new estimators of KL and in Section 3.6 we show how they can be used to improve the performance of E(BIC).

## 3.2 The GIC and the KLCV as estimators of the KL loss

In this section, we propose two new estimators of the KL in GGMs. Let $\widehat{\boldsymbol{\Theta}}_\lambda$ be the maximum penalized likelihood estimator defined by (2.8). The Kullback-Leibler divergence of the model $\mathcal{N}(\mathbf{0}, \widehat{\boldsymbol{\Theta}}_\lambda^{-1})$ from the true distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{\Theta}_0^{-1})$ can be written up to an additive constant as (see 2.7)

$$\mathrm{KL}(\boldsymbol{\Theta}_0; \widehat{\boldsymbol{\Theta}}_\lambda) = -\frac{1}{n} l(\widehat{\boldsymbol{\Theta}}_\lambda) + \mathrm{bias},$$

where $l(\boldsymbol{\Theta}) = n\{\log|\boldsymbol{\Theta}| - \mathrm{tr}(\boldsymbol{\Theta}\mathbf{S})\}/2$ and $\mathrm{bias} = \mathrm{tr}\{\widehat{\boldsymbol{\Theta}}_\lambda(\boldsymbol{\Theta}_0^{-1} - \mathbf{S})\}/2$. By estimating the bias term we obtain an estimate of the KL. The first estimator we propose, the so-called Generalized Information Criterion (GIC), has the form

$$\mathrm{GIC}(\lambda) = -2l(\widehat{\boldsymbol{\Theta}}_\lambda) + 2\widehat{\mathrm{df}}_{\mathrm{GIC}},$$

where

$$\widehat{\mathrm{df}}_{\mathrm{GIC}} = \frac{1}{2n}\sum_{k=1}^{n} \mathrm{vec}(\mathbf{S}_k \circ \mathbf{I}_\lambda)^\top \mathrm{vec}\{\widehat{\boldsymbol{\Theta}}_\lambda(\mathbf{S}_k \circ \mathbf{I}_\lambda)\widehat{\boldsymbol{\Theta}}_\lambda\} - \frac{1}{2}\mathrm{vec}(\mathbf{S} \circ \mathbf{I}_\lambda)^\top \mathrm{vec}\{\widehat{\boldsymbol{\Theta}}_\lambda(\mathbf{S} \circ \mathbf{I}_\lambda)\widehat{\boldsymbol{\Theta}}_\lambda\}, \tag{3.1}$$

where $\mathbf{I}_\lambda$ is the indicator matrix, whose entry is 1 if the corresponding entry in the precision matrix $\widehat{\boldsymbol{\Theta}}_\lambda$ is nonzero and zero if the corresponding entry in the precision matrix is zero. Here $\circ$ is the Schur or Hadamard product of matrices. In fact, GIC is an estimator of KL scaled by $2n$, which means that the estimator of the bias provided by GIC is $\widehat{\mathrm{df}}_{\mathrm{GIC}}/n$. We keep the scale in order to be consistent with the original definition of GIC (see 2.3).

Another estimator which we propose, referred to as the Kullback-Leibler cross-validation (KLCV), has the form

$$\mathrm{KLCV}(\lambda) = -\frac{1}{n} l(\widehat{\boldsymbol{\Theta}}_\lambda) + \widehat{\mathrm{bias}}_{\mathrm{KLCV}}, \tag{3.2}$$

where

$$\widehat{\mathrm{bias}}_{\mathrm{KLCV}} = \frac{1}{2n(n-1)}\sum_{k=1}^{n} \mathrm{vec}\{(\widehat{\boldsymbol{\Theta}}_\lambda^{-1} - \mathbf{S}_k) \circ \mathbf{I}_\lambda\}^\top \mathrm{vec}[\widehat{\boldsymbol{\Theta}}_\lambda\{(\mathbf{S} - \mathbf{S}_k) \circ \mathbf{I}_\lambda\}\widehat{\boldsymbol{\Theta}}_\lambda]. \tag{3.3}$$

To obtain the model $\mathcal{N}(\mathbf{0}, \widehat{\boldsymbol{\Theta}}_{\lambda^*})$, which is good in terms of prediction, we pick $\lambda^*$ that minimizes $\mathrm{GIC}(\lambda)$ or $\mathrm{KLCV}(\lambda)$ over $\lambda > 0$.

## 3.3 Derivation of the GIC and the KLCV

In this section we derive the GIC and the KLCV, in two steps. First, we derive the criteria for the maximum likelihood estimator. In the second step we use the derived formula and the assumption of sparsity to propose a formula for the maximum penalized likelihood estimator. The obtained formula is an extension of that for MLE, since both formulae are equivalent in the case of the maximum likelihood estimator.

### 3.3.1 Derivation of the GIC for the maximum likelihood estimator

The log-likelihood of one observation $\boldsymbol{y}_k$ for a Gaussian model $\mathcal{N}_p(\mathbf{0}, \boldsymbol{\Theta}^{-1})$ is, up to an additive constant, $l_k(\boldsymbol{\Theta}) = \frac{1}{2}\{\log|\boldsymbol{\Theta}| - \mathrm{tr}(\boldsymbol{\Theta}\mathbf{S}_k)\}$, where $\mathbf{S}_k = \boldsymbol{y}_k\boldsymbol{y}_k^\top$. The maxi-

mum likelihood estimator $\widehat{\boldsymbol{\Theta}}$ is an M-estimator defined as a solution of the system of equations

$$\sum_{k=1}^{n} \boldsymbol{\psi}(\boldsymbol{y}_k, \boldsymbol{\Theta}) = \mathbf{0}_{p^2}, \tag{3.4}$$

where $\boldsymbol{\psi}(\boldsymbol{y}_k, \boldsymbol{\Theta}) = \mathrm{vec}\mathsf{D}l_k(\boldsymbol{\Theta})$ and $\mathbf{0}_{p^2}$ is the $p^2$-dimensional zero vector. Consequently, the GIC for $\widehat{\boldsymbol{\Theta}}$ (see 2.3) is given by:

$$\mathrm{GIC} = -2l(\widehat{\boldsymbol{\Theta}}) + 2\mathrm{tr}(\mathbf{R}^{-1}\mathbf{Q}),$$

where $\mathbf{R}$ and $\mathbf{Q}$ are square matrices of order $p^2$

$$\mathbf{R} = -\frac{1}{n}\sum_{k=1}^{n}\{\mathsf{D}\boldsymbol{\psi}(\boldsymbol{y}_k, \widehat{\boldsymbol{\Theta}})\}^{\top}, \tag{3.5}$$

$$\mathbf{Q} = \frac{1}{n}\sum_{k=1}^{n}\boldsymbol{\psi}(\boldsymbol{y}_k, \widehat{\boldsymbol{\Theta}})\mathsf{D}l_k(\widehat{\boldsymbol{\Theta}}). \tag{3.6}$$

We are interested in deriving the term $\mathrm{tr}(\mathbf{R}^{-1}\mathbf{Q})$. Using matrix differential calculus (see Appendix 3.B) we obtain

$$\begin{aligned}\boldsymbol{\psi}(\boldsymbol{y}_k, \boldsymbol{\Theta}) = \mathrm{vec}\mathsf{D}l_k(\boldsymbol{\Theta}) &= \frac{1}{2}\mathrm{vec}(\boldsymbol{\Theta}^{-1} - \mathbf{S}_k),\\ \{\mathsf{D}\boldsymbol{\psi}(\boldsymbol{y}_k; \boldsymbol{\Theta})\}^{\top} &= -\frac{1}{2}\boldsymbol{\Theta}^{-1}\otimes\boldsymbol{\Theta}^{-1}.\end{aligned} \tag{3.7}$$

Using (3.4), (3.6) and equality $\widehat{\boldsymbol{\Theta}}^{-1} = \mathbf{S}$ (see Appendix 3.D for details) yields

$$\mathbf{Q} = \frac{1}{4n}\sum_{k=1}^{n}\mathrm{vec}\mathbf{S}_k\mathrm{vec}\mathbf{S}_k^{\top} - \frac{1}{4}\mathrm{vec}\mathbf{S}\mathrm{vec}\mathbf{S}^{\top}.$$

Expression for $\mathbf{R}^{-1}$ we obtain from (3.5) and (3.7)

$$\mathbf{R}^{-1} = 2\widehat{\boldsymbol{\Theta}}\otimes\widehat{\boldsymbol{\Theta}}.$$

Finally, formulas $\mathrm{tr}(\mathbf{AB}) = \mathrm{tr}(\mathbf{BA})$ and $\mathrm{tr}(\boldsymbol{x}^{\top}\mathbf{A}\boldsymbol{x}) = \boldsymbol{x}^{\top}\mathbf{A}\boldsymbol{x}$ imply that

$$\mathrm{tr}(\mathbf{R}^{-1}\mathbf{Q}) = \frac{1}{2n}\sum_{k=1}^{n}\mathrm{vec}\mathbf{S}_k^{\top}(\widehat{\boldsymbol{\Theta}}\otimes\widehat{\boldsymbol{\Theta}})\mathrm{vec}\mathbf{S}_k - \frac{1}{2}\mathrm{vec}\mathbf{S}^{\top}(\widehat{\boldsymbol{\Theta}}\otimes\widehat{\boldsymbol{\Theta}})\mathrm{vec}\mathbf{S}.$$

This formula is equivalent to (3.1) without the Schur product. This is shown in the Section 3.3.3.

### 3.3.2 Derivation of the KLCV for the maximum likelihood estimator

We follow the idea of Xiang and Wahba (1996), i.e. we introduce an approximation for LOOCV via several first order Taylor expansions. Lian (2011) uses this idea to derive the GACV for the MPLE in GGM, where in deriving the formula, the partial derivatives corresponding to the zero elements of the precision matrix are ignored. Here, unlike in Lian (2011), we apply the idea only for the MLE and therefore we avoid all technical difficulties entailed in ignoring the derivatives. We deal with the MPLE separately in the next section.

Consider the following function of two variables

$$f(\mathbf{S}, \mathbf{\Theta}) = \frac{2}{n} l(\mathbf{\Theta}) = \log|\mathbf{\Theta}| - \mathrm{tr}(\mathbf{S}\mathbf{\Theta}).$$

With this notation we have the identity

$$\sum_{k=1}^{n} f(\mathbf{S}_k, \mathbf{\Theta}) = n f(\mathbf{S}, \mathbf{\Theta}). \tag{3.8}$$

Let $\widehat{\mathbf{\Theta}}^{(-k)}$ be the MLE estimator of the precision matrix defined based on the data excluding the $k$th data point. The leave-one-out cross validation score (see 2.5 ) is defined by

$$\begin{aligned}
\text{LOOCV} &= -\frac{1}{n}\sum_{k=1}^{n} l_k(\widehat{\mathbf{\Theta}}^{(-k)}) = -\frac{1}{2n}\sum_{k=1}^{n} f(\mathbf{S}_k, \widehat{\mathbf{\Theta}}^{(-k)}) \\
&= -\frac{1}{2n}\sum_{k=1}^{n}\{f(\mathbf{S}_k, \widehat{\mathbf{\Theta}}^{(-k)}) - f(\mathbf{S}_k, \widehat{\mathbf{\Theta}}) + f(\mathbf{S}_k, \widehat{\mathbf{\Theta}})\} \\
&\overset{(3.8)}{=} -\frac{1}{2} f(\mathbf{S}, \widehat{\mathbf{\Theta}}) - \frac{1}{2n}\sum_{k=1}^{n}\{f(\mathbf{S}_k, \widehat{\mathbf{\Theta}}^{(-k)}) - f(\mathbf{S}_k, \widehat{\mathbf{\Theta}})\} \\
&\approx -\frac{1}{n} l(\widehat{\mathbf{\Theta}}) - \frac{1}{2n}\sum_{k=1}^{n}\left(\frac{\mathrm{d}f(\mathbf{S}_k, \widehat{\mathbf{\Theta}})}{\mathrm{d}\mathbf{\Theta}}\right)^{\top} \mathrm{vec}(\widehat{\mathbf{\Theta}}^{(-k)} - \widehat{\mathbf{\Theta}}).
\end{aligned}$$

Using matrix differential calculus (see Appendix 3.B) we deduce that $\mathrm{d}f(\mathbf{S}_k, \widehat{\mathbf{\Theta}})/\mathrm{d}\mathbf{\Theta} = \mathrm{vec}(\widehat{\mathbf{\Theta}}^{-1} - \mathbf{S}_k)^{\top}$. The term $\mathrm{vec}(\widehat{\mathbf{\Theta}}^{(-k)} - \widehat{\mathbf{\Theta}})$ is obtained via the Taylor expansion of the vector valued function $\left(\frac{\mathrm{d}f(\mathbf{S}^{(-k)}, \widehat{\mathbf{\Theta}}^{(-k)})}{\mathrm{d}\mathbf{\Theta}}\right)^{\top}$ around $(\mathbf{S}, \widehat{\mathbf{\Theta}})$. We expand the transposed

term because we consider vectors as columns.

$$\begin{aligned}\mathbf{0}_{p^2} &= \left(\frac{\mathrm{d}f(\mathbf{S}^{(-k)},\widehat{\boldsymbol{\Theta}}^{(-k)})}{\mathrm{d}\boldsymbol{\Theta}}\right)^\top \approx \left(\frac{\mathrm{d}f(\mathbf{S},\widehat{\boldsymbol{\Theta}})}{\mathrm{d}\boldsymbol{\Theta}}\right)^\top + \frac{\mathrm{d}^2 f(\mathbf{S},\widehat{\boldsymbol{\Theta}})}{\mathrm{d}\boldsymbol{\Theta}^2}\mathrm{vec}(\widehat{\boldsymbol{\Theta}}^{(-k)}-\widehat{\boldsymbol{\Theta}}) \\ &+ \frac{\mathrm{d}^2 f(\mathbf{S},\widehat{\boldsymbol{\Theta}})}{\mathrm{d}\boldsymbol{\Theta}\mathrm{d}\mathbf{S}}\mathrm{vec}(\mathbf{S}^{(-k)}-\mathbf{S}).\end{aligned}$$

From here it follows that

$$\mathrm{vec}(\widehat{\boldsymbol{\Theta}}^{(-k)}-\widehat{\boldsymbol{\Theta}}) \approx -\left(\frac{\mathrm{d}^2 f(\mathbf{S},\widehat{\boldsymbol{\Theta}})}{\mathrm{d}\boldsymbol{\Theta}^2}\right)^{-1}\frac{\mathrm{d}^2 f(\mathbf{S},\widehat{\boldsymbol{\Theta}})}{\mathrm{d}\boldsymbol{\Theta}\mathrm{d}\mathbf{S}}\mathrm{vec}(\mathbf{S}^{(-k)}-\mathbf{S}).$$

We have $\mathrm{d}f(\mathbf{S},\widehat{\boldsymbol{\Theta}})/\mathrm{d}\boldsymbol{\Theta} = \mathrm{vec}(\widehat{\boldsymbol{\Theta}}^{-1}-\mathbf{S})^\top$, so $\mathrm{d}^2 f(\mathbf{S},\widehat{\boldsymbol{\Theta}})/\mathrm{d}\boldsymbol{\Theta}\mathrm{d}\mathbf{S} = -\mathbf{I}_{p^2}$, $\mathrm{d}^2 f(\mathbf{S},\widehat{\boldsymbol{\Theta}})/\mathrm{d}\boldsymbol{\Theta}^2 = -\widehat{\boldsymbol{\Theta}}^{-1}\otimes\widehat{\boldsymbol{\Theta}}^{-1}$ and consequently

$$\mathrm{vec}(\widehat{\boldsymbol{\Theta}}^{(-k)}-\widehat{\boldsymbol{\Theta}}) \approx -(\widehat{\boldsymbol{\Theta}}\otimes\widehat{\boldsymbol{\Theta}})\mathrm{vec}(\mathbf{S}^{(-k)}-\mathbf{S}).$$

Therefore, the approximation of LOOCV, denoted by KLCV, has the form

$$\mathrm{KLCV} = -\frac{1}{n}l(\widehat{\boldsymbol{\Theta}}) + \frac{1}{2n}\sum_{k=1}^{n}\mathrm{vec}(\widehat{\boldsymbol{\Theta}}^{-1}-\mathbf{S}_k)^\top(\widehat{\boldsymbol{\Theta}}\otimes\widehat{\boldsymbol{\Theta}})\mathrm{vec}(\mathbf{S}^{(-k)}-\mathbf{S}).$$

After simplifying the term in the sum we finally obtain

$$\mathrm{KLCV} = -\frac{1}{n}l(\widehat{\boldsymbol{\Theta}}) + \frac{1}{2n(n-1)}\sum_{k=1}^{n}\mathrm{vec}(\widehat{\boldsymbol{\Theta}}^{-1}-\mathbf{S}_k)^\top(\widehat{\boldsymbol{\Theta}}\otimes\widehat{\boldsymbol{\Theta}})\mathrm{vec}(\mathbf{S}-\mathbf{S}_k).$$

This formula is equivalent to (3.2) without the Schur product. This is shown in the next section.

### 3.3.3 Extension of the GIC and the KLCV for the maximum penalized likelihood estimator

Before proposing the formulae for the penalized case we formulate two auxiliary results.

**Lemma 3.1.** *Let* $\mathbf{A}$ *and* $\boldsymbol{\Theta}$ *be symmetric matrices of order* $p$*. Then the following identity holds*

$$(\boldsymbol{\Theta}\otimes\boldsymbol{\Theta})\mathrm{vec}\mathbf{A} = \mathbf{M}_p(\boldsymbol{\Theta}\otimes\boldsymbol{\Theta})\mathrm{vec}\mathbf{A}. \tag{3.9}$$

**Lemma 3.2.** *Let* $\mathbf{A}$ *be a symmetric matrix of order* $p$ *and* $\boldsymbol{x}, \boldsymbol{y}$ *any vectors of dimension* $p$*. Then the value of the bilinear form*

$$B(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^\top \mathbf{A} \boldsymbol{y},$$

*when* $i$*th row (respectively column) of the matrix* $\mathbf{A}$ *is set to zero is the same as the value of* $B(\boldsymbol{x}, \boldsymbol{y})$ *when* $i$*th entry of the vector* $\boldsymbol{x}$ *(respectively* $\boldsymbol{y}$*) is set to zero.*

The proof of Lemma 3.1 is given in the Appendix 3.E; Lemma 3.2 is obtained by straightforward calculation. The penalty terms added to the log-likelihood in the GIC and the KLCV can be written in terms of the expression

$$T(\mathbf{A}, \mathbf{B}) = (\text{vec}\mathbf{A})^\top (\widehat{\boldsymbol{\Theta}} \otimes \widehat{\boldsymbol{\Theta}}) \text{vec}\mathbf{B}, \tag{3.10}$$

where $\mathbf{A}$ and $\mathbf{B}$ are symmetric matrices of order $p$. Indeed,

$$\text{bias}_{\text{KLCV}} = \frac{1}{2n(n-1)} \sum_{k=1}^{n} T(\widehat{\boldsymbol{\Theta}}^{-1} - \mathbf{S}_k, \mathbf{S} - \mathbf{S}_k), \tag{3.11}$$

$$\text{df}_{\text{GIC}} = \frac{1}{2n} \sum_{k=1}^{n} T(\mathbf{S}_k, \mathbf{S}_k) - \frac{1}{2} T(\mathbf{S}, \mathbf{S}). \tag{3.12}$$

According to Lemma 3.1 it follows that

$$T(\mathbf{A}, \mathbf{B}) = (\text{vec}\mathbf{A})^\top \mathbf{M}_p (\widehat{\boldsymbol{\Theta}} \otimes \widehat{\boldsymbol{\Theta}}) \text{vec}\mathbf{B}.$$

The fact that $2\mathbf{M}_p(\widehat{\boldsymbol{\Theta}} \otimes \widehat{\boldsymbol{\Theta}})$ is an estimator of the asymptotic covariance matrix of $\widehat{\boldsymbol{\Theta}}$ (Proposition 2.2) suggests an asymptotic argument for treating the case of a penalized estimator. Essentially, we only need to define the term $T(\mathbf{A}, \mathbf{B})$ for the MPLE.
To obtain the formula for the MPLE we assume standard conditions, as in Lam and Fan (2009), that guarantee a sparsistent MPLE. These conditions imply that $\lambda \to 0$ when $n \to \infty$, so we use formula (3.10), derived for the MLE, as an approximation in the penalized case. By sparsistency it follows that with probability one the zero coefficients are estimated as zero when $n$ tends to infinity. This implies that, asymptotically, the covariances between the zero elements and the nonzero elements in the estimated precision matrix are equal to zero. Thus, to obtain the term $T_\lambda(\mathbf{A}, \mathbf{B})$ for the MPLE we not only plug the expression $\widehat{\boldsymbol{\Theta}}_\lambda$ into the formula for the term $T(\mathbf{A}, \mathbf{B})$, but we also set the elements of the matrix $\mathbf{M}_p(\widehat{\boldsymbol{\Theta}}_\lambda \otimes \widehat{\boldsymbol{\Theta}}_\lambda)$ that correspond to covariances between the zero and nonzero elements of the precision matrix to zero. According to Lemma

3.2 this is equivalent to setting the corresponding entries of vectors vec$\mathbf{A}$ and vec$\mathbf{B}$ to zero, i.e.

$$\begin{aligned} T_\lambda(\mathbf{A},\mathbf{B}) &\stackrel{\text{def}}{=} \operatorname{vec}(\mathbf{A}\circ\mathbf{I}_\lambda)^\top \mathbf{M}_p(\widehat{\boldsymbol{\Theta}}_\lambda \otimes \widehat{\boldsymbol{\Theta}}_\lambda)\operatorname{vec}(\mathbf{B}\circ\mathbf{I}_\lambda) \\ &= \operatorname{vec}(\mathbf{A}\circ\mathbf{I}_\lambda)^\top (\widehat{\boldsymbol{\Theta}}_\lambda \otimes \widehat{\boldsymbol{\Theta}}_\lambda)\operatorname{vec}(\mathbf{B}\circ\mathbf{I}_\lambda), \end{aligned} \tag{3.13}$$

where the second equality follows from Lemma 3.1. The obtained formula involves matrices of order $p^2$, which entails high cost in terms of memory usage and floating-point operations. For this reason, we rewrite the formula in a way that it is computationally feasible. Following Lian (2011) we apply identity $\operatorname{vec}(\mathbf{ABC}) = (\mathbf{C}^\top \otimes \mathbf{A})\operatorname{vec}\mathbf{B}$ to (3.13) and obtain

$$T_\lambda(\mathbf{A},\mathbf{B}) = \operatorname{vec}(\mathbf{A}\circ\mathbf{I}_\lambda)^\top \operatorname{vec}\{\widehat{\boldsymbol{\Theta}}_\lambda(\mathbf{B}\circ\mathbf{I}_\lambda)\widehat{\boldsymbol{\Theta}}_\lambda\}. \tag{3.14}$$

For details on the implementation of this formula, see the next section. Finally, it follows that bias and degrees of freedom terms for the GIC and the KLCV in the case of the MPLE are obtained by substituting $T$ with $T_\lambda$ in expressions (3.11) and (3.12), i.e

$$\begin{aligned} \operatorname{bias}_{\text{KLCV}}(\lambda) &= \frac{1}{n(n-1)} \sum_{k=1}^{n} T_\lambda(\widehat{\boldsymbol{\Theta}}_\lambda^{-1} - \mathbf{S}_k, \mathbf{S} - \mathbf{S}_k), \\ \operatorname{df}_{\text{GIC}}(\lambda) &= \frac{1}{2n} \sum_{k=1}^{n} T_\lambda(\mathbf{S}_k, \mathbf{S}_k) - \frac{1}{2} T(\mathbf{S}, \mathbf{S}), \end{aligned}$$

which are formulae presented in (3.3) and (3.1).

To conclude this section, we show that the derived formulae for the MPLE are extensions of the corresponding formulae for the MLE, meaning that applying the MPLE formulae to the maximum likelihood estimator yields the same result as the corresponding MLE formulae. To this aim, let $\widehat{\boldsymbol{\Theta}}$ be the maximum likelihood estimator of the precision matrix, which is the MPLE for $\lambda = 0$, i.e. $\widehat{\boldsymbol{\Theta}} = \widehat{\boldsymbol{\Theta}}_\lambda$, for $\lambda = 0$. Since with probability one all the elements of $\widehat{\boldsymbol{\Theta}}$ are nonzero it follows that $\mathbf{I}_\lambda$ is the matrix with all entries equal to one. This implies that in the formula (3.13) we have $\mathbf{A}\circ\mathbf{I}_\lambda = \mathbf{A}$ and $\mathbf{B}\circ\mathbf{I}_\lambda = \mathbf{B}$, which in turn implies $T_\lambda(\mathbf{A},\mathbf{B}) = T(\mathbf{A},\mathbf{B})$.

## 3.4 Implementation

In this section, we give more details as to why formula (3.14) is computationally much more efficient than (3.13). Also, we show how to implement (3.14) efficiently. First, the computational complexity of (3.14) is $\mathcal{O}(p^3)$ while for (3.13) it is $\mathcal{O}(p^4)$ (see Appendix 3.F for details). Furthermore, (3.13) requires storing matrices of order $p^2$, which is $p^4$ elements per matrix. Even for moderate $p$, allocation of that amount of memory on a standard desktop computer is not possible. On the other hand, the formula (3.14) is written in terms of vectors of dimension $p^2$, which implies storing of $p^2$ elements per vector. To further simplify, we demonstrate that we can avoid the usage of the transpose and vectorization operators. For any matrices $\mathbf{X} = (x_{ij})$ and $\mathbf{Y} = (y_{ij})$ it holds that $(\mathrm{vec}\mathbf{X})^\top \mathrm{vec}\mathbf{Y} = \sum_{i,j} x_{ij} y_{ij}$ so it follows that $(\mathrm{vec}\mathbf{X})^\top \mathrm{vec}\mathbf{Y}$ is just the sum of the elements of the matrix $\mathbf{X} \circ \mathbf{Y}$, i.e. $(\mathrm{vec}\mathbf{X})^\top \mathrm{vec}\mathbf{Y} = \sum_{i,j} (\mathbf{X} \circ \mathbf{Y})_{ij}$. Applying this to (3.14) we obtain

$$T_\lambda(\mathbf{A}, \mathbf{B}) = \sum_{i,j} (\mathbf{A} \circ \mathbf{I}_\lambda \circ \{\widehat{\mathbf{\Theta}}_\lambda (\mathbf{B} \circ \mathbf{I}_\lambda) \widehat{\mathbf{\Theta}}_\lambda\})_{ij}.$$

In the statistical programming language R, expression $\sum_{i,j} (\mathbf{X} \circ \mathbf{Y})_{ij}$ can be efficiently implemented as `sum(X*Y)`. This can be used in expressions (3.1) and (3.3).

## 3.5 Simulation study

In this section, we test the performance of the proposed formulas in terms of Kullback-Leibler loss. We do this in the case of the LASSO penalty for two sparse hub graphs. The graphs have $p = 40$ nodes and 38 edges and $p = 100$ nodes and 95 edges. The sparsity values of these graphs are 0.049 and 0.019 respectively. The graphs are shown in Figure 3.1. We omit the results for other type of graphs having the same combinations of $n$ and $p$. These methods were tested for a band graph, a random graph, a cluster graph and a scale-free graph. Our estimators exhibits similar performance in all these cases. All the simulations are done on a grid of two hundred regularization parameters.

We compare the following estimators: the KL oracle estimator, the proposed KLCV estimator, and the AIC and the GACV estimators. The KL oracle estimator is that $\mathbf{\Theta}_\lambda$ in the LASSO solution path that minimizes the KL loss if we know the true matrix. Under each model we have generated 100 simulated data sets with different

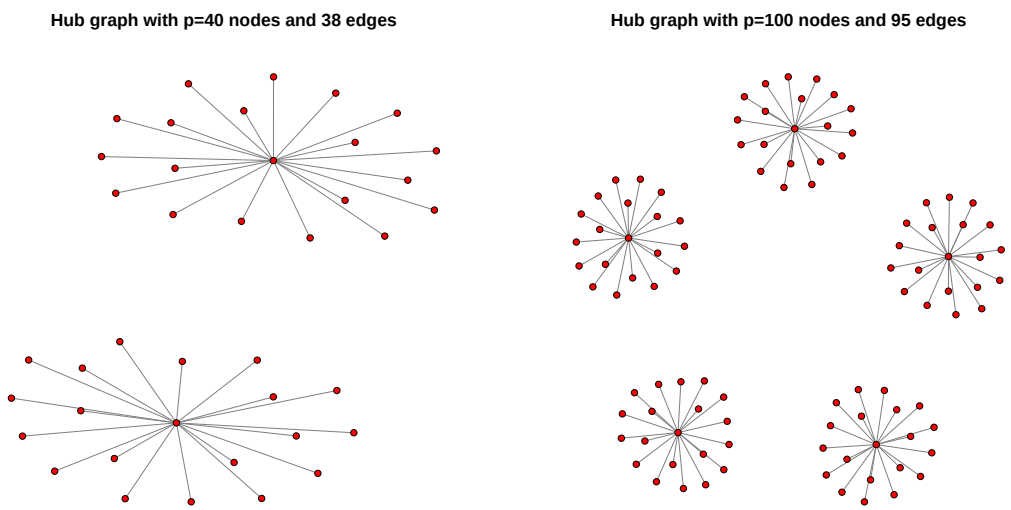



Fig. 3.1 Hub graphs with $p = 40$ and $p = 100$ nodes used in the simulation study.

combinations of $p$ and $n$. We focus on cases where $n \leq p$, which are more common in the applications. For the simulations we use the **huge** package in R (Zhao et al., 2014). The results are given in Tables 3.1 and 3.2. The KLCV outperforms the AIC and the GACV for all sample sizes when $p = 40$ and for $p = 100$. For $n = 400$ the KLCV is slightly worse than the AIC. The KLCV method is close to the KL oracle score, even for very small $n$. Overall, the KLCV exhibits performance similar to that of AIC and the GACV in large sample size scenarios, but it clearly outperforms both when the sample size is small. The GIC also outperforms the AIC and the GACV, except that it underperforms in comparison to the AIC when the sample size is very small; these results are seen in cases $n = 8, 12$ for $p = 40$ and $n = 20, 30$ for $p = 100$. As for the KLCV and the GIC, results show that the KLCV is overall better and more stable. The KLCV performs well for any sample size whereas the GIC is less adequate for very small sample sizes.

Computationally, our formulae are slightly slower than that of the GACV since we have an additional Schur product in the calculation.

| p=40 | KL ORACLE | KLCV | GIC | AIC | GACV |
|---|---|---|---|---|---|
| n=8 | 3.68 | **3.71** | 24.11 | 6.46 | 26.80 |
| | (0.27) | **(0.28)** | (2.92) | (2.12) | (1.66) |
| n=12 | 3.29 | **3.36** | 9.28 | 6.58 | 18.34 |
| | (0.26) | **(0.28)** | (2.28) | (3.54) | (1.61) |
| n=16 | 2.93 | **3.01** | 5.11 | 6.62 | 13.07 |
| | (0.26) | **(0.26)** | (1.09) | (3.07) | (1.36) |
| n=20 | 2.67 | **2.76** | 3.74 | 6.48 | 10.08 |
| | (0.23) | **(0.25)** | (0.62) | (2.50) | (1.20) |
| n=30 | 2.18 | **2.27** | 2.5 | 4.59 | 5.81 |
| | (0.23) | **(0.25)** | (0.37) | (1.11) | (0.66) |
| n=40 | 1.91 | **2.00** | 2.05 | 3.18 | 4.13 |
| | (0.19) | **(0.21)** | (0.21) | (0.66) | (0.43) |
| n=100 | 1.00 | 1.04 | **1.02** | 1.17 | 1.32 |
| | (0.10) | (0.11) | **(0.11)** | (0.11) | (0.14) |

Table 3.1 Simulation results for hub graph with $p = 40$ nodes. Performance in terms of Kullback-Leibler loss of different estimators for different sample size $n$ is shown. The results are based on 100 simulated data sets. Standard errors are shown in brackets. The best result is boldfaced and the second best is underlined.

## 3.6 Using the KLCV and the GIC for graph estimation

Information criteria, such as AIC, (E)BIC, for model selection in Gaussian graphical models are based on penalizing the log-likelihood with a term that involves the degrees of freedom, which are defined as

$$\mathrm{df}(\lambda) = \sum_{1 \le i < j \le p} I(\widehat{\theta}_{ij,\lambda} \neq 0), \tag{3.15}$$

where $(\widehat{\theta}_{ij,\lambda})_{1 \le i < j \le p}$ are the estimated parameters (Yuan and Lin, 2007). As we pointed out in Section 3.1 not the AIC should be used for graph estimation but (E)BIC. However, even though BIC and EBIC have the consistency property, in a sparse data setting they can perform poorly because of the instability of the degrees of freedom defined in (3.15). As Li and Gui (2006) point out, in high-dimensional cases there is often considerable uncertainty in the number of non-zero elements in the precision matrix. To overcome this uncertainty, the authors propose to use the bootstrap method to determine the statistical accuracy and the importance of each non-zero el-

| p=100 | KL ORACLE | KLCV | GIC | AIC | GACV |
|---|---|---|---|---|---|
| n=20 | 8.06 | **8.60** | 29.7 | 12.24 | 28.59 |
| | (0.37) | **(0.45)** | (4.56) | (0.28) | (19.94) |
| n=30 | 6.87 | **7.29** | 11.5 | 10.59 | 32.07 |
| | (0.34) | **(0.39)** | (1.82) | (0.41) | (2.77) |
| n=40 | 5.92 | **6.34** | 7.5 | 9.15 | 22.48 |
| | (0.30) | **(0.38)** | (0.84) | (0.59) | (1.88) |
| n=50 | 5.24 | **5.63** | 5.99 | 7.33 | 16.93 |
| | (0.27) | **(0.33)** | (0.59) | (0.81) | (1.40) |
| n=75 | 4.08 | 4.36 | **4.26** | 4.76 | 9.80 |
| | (0.27) | (0.31) | **(0.31)** | (0.71) | (0.71) |
| n=100 | 3.34 | 3.57 | **3.4** | 3.63 | 6.81 |
| | (0.19) | (0.23) | **(0.19)** | (0.48) | (0.52) |
| n=400 | 1.13 | 1.20 | **1.17** | **1.17** | 1.24 |
| | (0.07) | (0.08) | **(0.08)** | **(0.08)** | (0.07 ) |

Table 3.2 Simulation results for hub graph with $p = 40$ nodes. Performance in terms of Kullback-Leibler loss of different estimators for different sample size $n$ is showed. The results are based on 100 simulated data sets. Standard errors are shown in brackets. The best result is boldfaced and the second best is underlined.

ement identified by the proposed procedure. One can then choose only those elements with high probability of being non-zero in the precision matrix across the bootstrap samples. Here we propose an alternative, faster, approach.

Recall that AIC has the form

$$\mathrm{AIC}(\lambda) = -2l(\widehat{\mathbf{\Theta}}_\lambda) + 2\mathrm{df}(\lambda),$$

where $\mathrm{df}(\lambda)$ is defined in (3.15). The AIC is an estimator of KL loss scaled by $2n$. It follows that the degrees of freedom in the AIC provide an estimator of the bias of the KL loss scaled by $n/2$. Since, the KLCV also provides an estimator of this bias, we define

$$\mathrm{df}_{\mathrm{KLCV}}(\lambda) = \frac{n}{2}\widehat{\mathrm{bias}}_{\mathrm{KLCV}},$$

where $\widehat{\mathrm{bias}}_{\mathrm{KLCV}}$ is defined in (3.3). For the GIC no scaling is needed, since the degrees of freedom are already defined in (3.1). Since the (E)BIC has better graph selection properties than the AIC, we can use $\mathrm{df}_{\mathrm{KLCV}}(\lambda)$ and $\mathrm{df}_{\mathrm{GIC}}(\lambda)$ with the (E)BIC. In

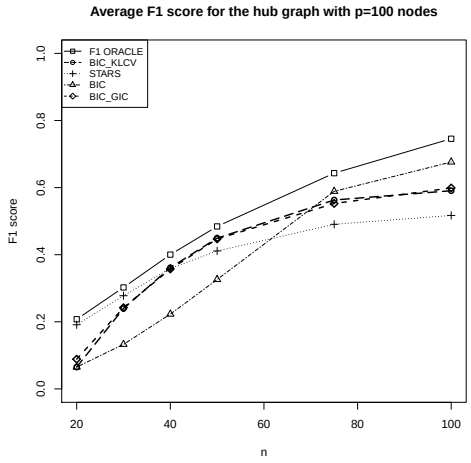
other words, we define

$$\mathrm{BIC}_{\mathrm{KLCV}}(\lambda) = -2l(\widehat{\boldsymbol{\Theta}}_\lambda) + \log n \mathrm{df}_{\mathrm{KLCV}}(\lambda),$$
$$\mathrm{BIC}_{\mathrm{GIC}}(\lambda) = -2l(\widehat{\boldsymbol{\Theta}}_\lambda) + \log n \mathrm{df}_{\mathrm{GIC}}(\lambda).$$

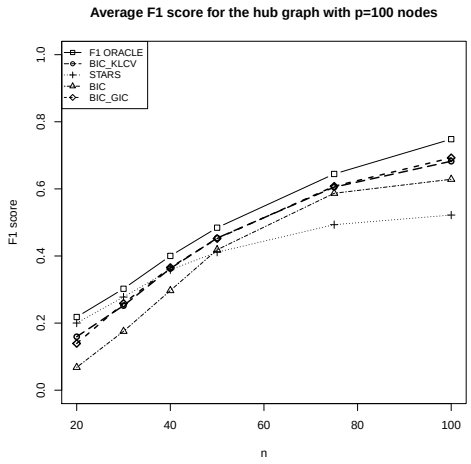
We can also do the same for EBIC. We compare the $\mathrm{BIC}_{\mathrm{GIC}}(\lambda)$ and $\mathrm{BIC}_{\mathrm{KLCV}}(\lambda)$ to BIC and StARS in terms of the $\mathrm{F}_1$ score. The largest possible value of the $\mathrm{F}_1$ score is given by the $\mathrm{F}_1$ oracle and is evaluated by using the true matrix. We have done simulations for the graph with $p = 100$ nodes from the previous section and a grid of two hundred regularization parameters. Average results over 100 simulations are given in Figure 3.2. The results suggest that $\mathrm{BIC}_{\mathrm{GIC}}(\lambda)$ and $\mathrm{BIC}_{\mathrm{KLCV}}(\lambda)$ can improve BIC for small sample sizes and can be competitive with the computationally much more involved StARS. In the case of the adaptive LASSO penalty the improvement is evident. Theoretical properties of $\mathrm{BIC}_{\mathrm{KLCV}}(\lambda)$ and $\mathrm{BIC}_{\mathrm{GIC}}(\lambda)$ are unclear, due to the complicated form of the bias terms. In any case, we propose the method only when the sample size is small since for larger $n$ the degrees of freedom defined in 3.15 are more stable, which can be also seen from the performance of AIC in Tables 3.1 and 3.2.

## 3.7 Summary

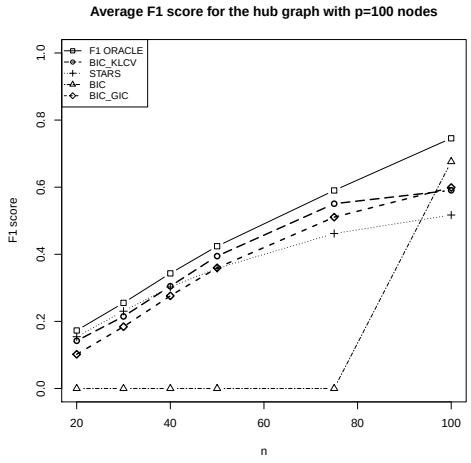
In this chapter, we have discussed model selection in Gaussian graphical models. We argued that minimizing KL divergence performs well in obtaining a model with good predicting power, but poorly in retrieving the graph structure. For obtaining a model with a good predictive power we have proposed two new, closed-form estimators of the Kullback-Leibler loss. After an extensive literature review, we have found that these estimators are the best closed-form estimators available for selecting a predictively accurate model in sparse data settings for sparse Gaussian graphical models. We have demonstrated that the estimators can be implemented relatively efficiently. We have concluded the chapter by illustrating that the proposed estimators of KL can be useful for the graph selection problem when the sample size is small.




(a) GLASSO




(b) SCAD




(c) ADAPTIVE GLASSO

Fig. 3.2 Simulations results for hub graph with $p = 100$ nodes. Average performance in terms of $F_1$ score of different estimators for different sample size $n$ is shown. The results are based on 100 simulated data sets.

## 3.A Proof of Theorem 3.1

The proof follows the line of reasoning in the proof of Theorem 1 given in (Zhang et al., 2010). First we list the necessary notation:

- $\bar{\alpha} = \{(i,j) : i,j = 1,\dots,p;\ i < j\}$ is the full model (labels of all entries in the upper diagonal of the precision matrix ).
- $\alpha \subset \bar{\alpha}$ is the candidate model which is defined by the set of labels of nonzero entries of the precision matrix $\alpha$.
- $\mathcal{A}$ is the collection of all candidate models.
- $\widehat{\mathbf{\Theta}}^*_\alpha$ is the nonpenalized MLE computed via the model $\alpha$.
- $\widehat{\mathbf{\Theta}}_\lambda$ is the MPLE that corresponds to tuning parameter $\lambda$.
- $\alpha_\lambda$ is the model associated with $\widehat{\mathbf{\Theta}}_\lambda$.
- $d_\alpha$ is the size of the model $\alpha$, i.e. the number of nonzero elements in the upper diagonal of the precision matrix $\widehat{\mathbf{\Theta}}^*_\alpha$.
- $df_\lambda$ is the size of the model $\alpha_\lambda$, i.e. the number of nonzero elements in the upper diagonal of the precision matrix $\widehat{\mathbf{\Theta}}_\lambda$.
- $\Omega_- = \{\lambda : \alpha_\lambda \not\supset \alpha_0\}$ is the set of $\lambda$s that induce an underfitted model.
- $\Omega_0 = \{\lambda : \alpha_\lambda = \alpha_0\}$ is the set of $\lambda$s that induce the true model.
- $\Omega_+ = \{\lambda : \alpha_\lambda \supset \alpha_0 \text{ and } \alpha_\lambda \neq \alpha_0\}$ is the set of $\lambda$s that induce an overfitted model.
- $l(\mathbf{\Theta}) = \frac{n}{2}\{\log|\mathbf{\Theta}| - \mathrm{tr}(\mathbf{\Theta S})\}$ is the log-likelihood.
- $D(\lambda) = -2l(\widehat{\mathbf{\Theta}}_\lambda)$ is the deviance evaluated in the MPLE $\widehat{\mathbf{\Theta}}_\lambda$.
- $D^*(\alpha) = -2l(\widehat{\mathbf{\Theta}}^*_\alpha)$ is the deviance evaluated in the MLE $\widehat{\mathbf{\Theta}}^*_\alpha$ that corresponds to model $\alpha$.
- $\mathrm{AIC}(\lambda) = D(\lambda)/n + 2df_\lambda/n$ is the AIC for the MPLE $\widehat{\mathbf{\Theta}}_\lambda$.
- $\mathrm{AIC}^*(\alpha) = D^*(\alpha)/n + 2d_\alpha/n$ is the AIC for the MLE $\widehat{\mathbf{\Theta}}^*_\alpha$ that corresponds to model $\alpha$.

There are two results needed to prove the statement of the theorem; these are given in lemmas 3.3 and 3.4. Then, application of the apparatus of (Zhang et al., 2010) to our framework yields lemmas 3.5 and 3.6 from which follows the statement of the theorem.

**Lemma 3.3.** *For any candidate model $\alpha \in \mathcal{A}$, there exists $c_\alpha$ such that*

$$\frac{D^*(\alpha)}{n} = c_\alpha + o_P(1).$$

*In addition, for any underfitted model $\alpha \not\supset \alpha_0$, it holds*

$$c_\alpha > c_{\alpha_0},$$

*where $\widehat{\boldsymbol{\Theta}}^*_{\alpha_0}$ is the precision matrix of the true model $\alpha_0$.*

*Proof.* From the definition of $D^*(\alpha)$ it follows that

$$\frac{D^*(\alpha)}{n} = \operatorname{tr}(\widehat{\boldsymbol{\Theta}}^*_\alpha \mathbf{S}) - \log|\widehat{\boldsymbol{\Theta}}^*_\alpha|.$$

Fix any candidate model $\alpha \in \mathcal{A}$. Since $\widehat{\boldsymbol{\Theta}}^*_\alpha$ is the MLE under the model $\alpha$ the limit $\lim_{n\to\infty} \widehat{\boldsymbol{\Theta}}^*_\alpha$ exists in probability; we denote it by $\boldsymbol{\Theta}_\alpha$. By consistency of the estimator $\mathbf{S}$ (MLE under the full model) it follows that $\mathbf{S} \xrightarrow{P} \boldsymbol{\Theta}^{-1}_{\alpha_0}$, where $\boldsymbol{\Theta}_{\alpha_0}$ is the true precision matrix. According to the continuous mapping theorem it follows that in probability it holds that

$$\lim_{n\to\infty} \frac{D^*(\alpha)}{n} = \operatorname{tr}(\boldsymbol{\Theta}_\alpha \boldsymbol{\Theta}^{-1}_{\alpha_0}) - \log|\boldsymbol{\Theta}_\alpha| := c_\alpha, \tag{3.16}$$

which establishes the existence of $c_\alpha$. Let $\alpha \not\supset \alpha_0$ be any underfitted model. According to (3.16)

$$\begin{aligned} c_\alpha - c_{\alpha_0} &= \operatorname{tr}(\boldsymbol{\Theta}_\alpha \boldsymbol{\Theta}^{-1}_{\alpha_0}) - \log|\boldsymbol{\Theta}_\alpha| - \operatorname{tr}(\boldsymbol{\Theta}_{\alpha_0} \boldsymbol{\Theta}^{-1}_{\alpha_0}) + \log|\boldsymbol{\Theta}_{\alpha_0}| \\ &= \operatorname{tr}(\boldsymbol{\Theta}_\alpha \boldsymbol{\Theta}^{-1}_{\alpha_0}) - \log|\boldsymbol{\Theta}_\alpha \boldsymbol{\Theta}^{-1}_{\alpha_0}| - p \\ &= 2\mathrm{KL}(\boldsymbol{\Theta}_{\alpha_0}; \boldsymbol{\Theta}_\alpha) > 0, \end{aligned}$$

where we have used the formula for KL (2.6), the proposition 2.3 and the fact that the densities $\mathcal{N}_p(\boldsymbol{y}; 0, \boldsymbol{\Theta}^{-1}_\alpha)$ are not equal $\mathcal{N}_p(\boldsymbol{y}; 0, \boldsymbol{\Theta}^{-1}_{\alpha_0})$. □

**Lemma 3.4.** *Let $\alpha_0$ denote the true model and $\bar{\alpha}$ the full model. With the notation of the previous lemma we have $c_{\alpha_0} = c_{\bar{\alpha}}$.*

*Proof.* Let $\boldsymbol{\Theta}_{\alpha_0}$ be the true precision matrix. According to (3.16)

$$c_\alpha = \operatorname{tr}(\boldsymbol{\Theta}_\alpha \boldsymbol{\Theta}^{-1}_{\alpha_0}) - \log|\boldsymbol{\Theta}_\alpha|,$$

where $\boldsymbol{\Theta}_\alpha = \lim_{n\to\infty} \widehat{\boldsymbol{\Theta}}^*_\alpha$, in probability. It follows that

$$c_{\alpha_0} = \operatorname{tr}(\boldsymbol{\Theta}_{\alpha_0} \boldsymbol{\Theta}^{-1}_{\alpha_0}) - \log|\boldsymbol{\Theta}_{\alpha_0}| = p - \log|\boldsymbol{\Theta}_{\alpha_0}|.$$

From $\widehat{\boldsymbol{\Theta}}^*_{\bar{\alpha}} = \mathbf{S}$ it follows that $\boldsymbol{\Theta}_{\bar{\alpha}} = \boldsymbol{\Theta}_{\alpha_0}$, because $\mathbf{S}$ is a consistent estimator.

Finally we obtain

$$c_{\bar{\alpha}} = \operatorname{tr}(\boldsymbol{\Theta}_{\bar{\alpha}}\boldsymbol{\Theta}_{\alpha_0}^{-1}) - \log|\boldsymbol{\Theta}_{\bar{\alpha}}| = \operatorname{tr}(\boldsymbol{\Theta}_{\alpha_0}\boldsymbol{\Theta}_{\alpha_0}^{-1}) - \log|\boldsymbol{\Theta}_{\alpha_0}| = p - \log|\boldsymbol{\Theta}_{\alpha_0}| = c_{\alpha_0}.$$

□

**Lemma 3.5.** *$P\left\{\inf_{\lambda\in\Omega_-} \mathrm{AIC}(\lambda) > \mathrm{AIC}^*(\bar{\alpha})\right\} \to 1$, as $n \to \infty$.*

*Proof.* For a given $\lambda > 0$, the nonpenalized MLE $\widehat{\boldsymbol{\Theta}}^*_{\alpha_\lambda}$ maximizes log-likelihood function under the model $\alpha_\lambda$. Therefore, $l(\widehat{\boldsymbol{\Theta}}^*_{\alpha_\lambda}) \geq l(\widehat{\boldsymbol{\Theta}}_\lambda)$ which is equivalent to

$$D(\lambda) \geq D^*(\alpha_\lambda).$$

According to previous inequality and definition of the $\mathrm{AIC}(\lambda)$ it follows that

$$\mathrm{AIC}(\lambda) = \frac{D(\lambda)}{n} + \frac{2df_\lambda}{n} > \frac{D^*(\alpha_\lambda)}{n} + \frac{2df_\lambda}{n} > \frac{D^*(\alpha_\lambda)}{n},$$

since $2df_\lambda/n > 0$. Hence

$$\mathrm{AIC}(\lambda) - \mathrm{AIC}^*(\bar{\alpha}) > \frac{D^*(\alpha_\lambda)}{n} - \frac{D^*(\bar{\alpha})}{n} - \frac{2d_{\bar{\alpha}}}{n}, \text{ for every } \lambda > 0. \tag{3.17}$$

Consequently, the last inequality also holds for any $\lambda \in \Omega_- = \{\lambda : \alpha_\lambda \not\supset \alpha_0\}$, which implies

$$P\left\{\inf_{\lambda\in\Omega_-} \mathrm{AIC}(\lambda) - \mathrm{AIC}^*(\bar{\alpha}) > 0\right\} \geq P\left\{\inf_{\alpha_\lambda \not\supset \alpha_0} \frac{D^*(\alpha_\lambda)}{n} - \frac{D^*(\bar{\alpha})}{n} - \frac{2d_{\bar{\alpha}}}{n} > 0\right\}. \tag{3.18}$$

Since the subset of all underfitted models contains the subset of underfitted models obtained by the MPLE, i.e. $\{\alpha_\lambda : \alpha_\lambda \not\supset \alpha_0\} \subset \{\alpha : \alpha \not\supset \alpha_0\}$ it follows that

$$P\left\{\inf_{\alpha_\lambda \not\supset \alpha_0} \frac{D^*(\alpha_\lambda)}{n} - \frac{D^*(\bar{\alpha})}{n} - \frac{2d_{\bar{\alpha}}}{n} > 0\right\} \geq P\left\{\inf_{\alpha \not\supset \alpha_0} \frac{D^*(\alpha)}{n} - \frac{D^*(\bar{\alpha})}{n} - \frac{2d_{\bar{\alpha}}}{n} > 0\right\}. \tag{3.19}$$

Because $p$ is fixed and it holds that $0 \leq \frac{2d_{\bar{\alpha}}}{n} \leq \frac{2p^2}{n}$ we obtain that $\frac{2d_{\bar{\alpha}}}{n} = o_P(1)$. Lemma 3.3 implies that $D^*(\alpha)/n - D^*(\bar{\alpha})/n - 2d_{\bar{\alpha}}/n = c_\alpha - c_{\bar{\alpha}} + o_P(1)$, whence it follows that

$$P\left\{\inf_{\alpha \not\supset \alpha_0} \frac{D^*(\alpha)}{n} - \frac{D^*(\bar{\alpha})}{n} - \frac{2d_{\bar{\alpha}}}{n} > 0\right\} = P\left\{\inf_{\alpha \not\supset \alpha_0} c_\alpha - c_{\bar{\alpha}} + o_P(1) > 0\right\}. \tag{3.20}$$

Since there are a finite number of underfitted models, infimum is equal to minimum so

$$P\left\{\inf_{\alpha\not\supset\alpha_0} c_\alpha - c_{\bar{\alpha}} + o_P(1) > 0\right\} = P\left\{\min_{\alpha\not\supset\alpha_0} c_\alpha - c_{\bar{\alpha}} + o_P(1) > 0\right\}. \tag{3.21}$$

According to Lemma 3.4 $c_{\bar{\alpha}} = c_{\alpha_0}$. From this equality, the sequence of inequalities (3.18), (3.19), (3.20), (3.21) and Lemma 3.3, i.e. $c_\alpha > c_{\alpha_0}$, it follows that

$$P\left\{\inf_{\lambda\in\Omega_-} \mathrm{AIC}(\lambda) > \mathrm{AIC}^*(\alpha)\right\} \to 1 \quad \text{as} \quad n\to\infty.$$

□

**Lemma 3.6.** $\liminf_{n\to\infty} P\left\{\inf_{\lambda\in\Omega_0} \mathrm{AIC}(\lambda) > \mathrm{AIC}^*(\bar{\alpha})\right\} \geq \pi > 0.$

*Proof.* Let $\lambda\in\Omega_0$ be arbitrary. Applying (3.17) to $\alpha_\lambda = \alpha_0$ and taking infimum over $\lambda\in\Omega_0$ yields

$$\begin{aligned}\inf_{\lambda\in\Omega_0} \mathrm{AIC}(\lambda) - \mathrm{AIC}^*(\bar{\alpha}) &> \frac{D^*(\alpha_0)}{n} - \frac{D^*(\bar{\alpha})}{n} - \frac{2d_{\bar{\alpha}}}{n} \\ &= \frac{1}{n}\left\{D^*(\alpha_0) - D^*(\bar{\alpha}\right\} - \frac{2d_{\bar{\alpha}}}{n}. \end{aligned}\tag{3.22}$$

According to (Lauritzen, 1996, p.142)

$$D^*(\alpha_0) - D^*(\bar{\alpha}) = -2\{l(\widehat{\boldsymbol{\Theta}}^*_{\alpha_0}) - l(\widehat{\boldsymbol{\Theta}}^*_{\bar{\alpha}})\} \xrightarrow{d} \chi^2_{d_{\bar{\alpha}}-d_{\alpha_0}}, \tag{3.23}$$

where $\chi^2_d$ denotes the chi-squared distribution with $d$ degrees of freedom. Finally (3.22) and (3.23) imply

$$\begin{aligned} P\left\{\inf_{\lambda\in\Omega_0} \mathrm{AIC}(\lambda) > \mathrm{AIC}^*(\bar{\alpha})\right\} &\geq P\left\{-\frac{2}{n}\{l(\widehat{\boldsymbol{\Theta}}^*_{\alpha_0}) - l(\widehat{\boldsymbol{\Theta}}^*_{\bar{\alpha}})\} - \frac{2d_{\bar{\alpha}}}{n} > 0\right\} \\ &= P\left\{-2\{l(\widehat{\boldsymbol{\Theta}}^*_{\alpha_0}) - l(\widehat{\boldsymbol{\Theta}}^*_{\bar{\alpha}})\} > 2d_{\bar{\alpha}}\right\} \\ &\to P\{\chi^2_{d_{\bar{\alpha}}-d_{\alpha_0}} > 2d_{\bar{\alpha}}\} = \pi, \end{aligned}$$

where $0<\pi<1$. □

## 3.B Matrix differential calculus

If $\mathbf{F}$ is a differentiable $m\times p$ matrix function of an $n\times q$ matrix $\mathbf{X}$ of variables then the natural question is how to define the Jacobian matrix of $\mathbf{F}$. The literature gives

different definitions, since it is possible to construct the matrix containing the $mnpq$ partial derivatives of $\mathbf{F}$ in many ways. In this thesis we opt for the definition proposed in Magnus and Neudecker (1985), where the authors point out the following issues with other definitions of the derivative.

1. The definition does not give the Jacobian matrix, it just displays the partial derivatives.
2. The determinant of the matrix that contains partial derivatives has no interpretation.
3. The definition is such that the chain rule does not exist.

The authors show that the only natural and viable generalization of the notion of a Jacobian matrix of a vector function to a Jacobian matrix of a matrix function is the one that we present here.

**Definition 3.1.** *Let $\phi$ be a scalar function of an $n \times 1$ vector $\boldsymbol{x} = (x_1, \ldots, x_n)^\top$, $\boldsymbol{f}$ be an $m \times 1$ vector function of $\boldsymbol{x}$ and $\mathbf{F}$ be a differentiable $m \times p$ real matrix function of an $n \times q$ matrix of real variables $\mathbf{X} = (x_{ij})$. Then the derivative of $\phi$ at $\boldsymbol{x}$ is the $n \times 1$ vector*

$$\mathsf{D}\phi(\boldsymbol{x}) = \left( \frac{\partial \phi(\boldsymbol{x})}{\partial x_1}, \ldots, \frac{\partial \phi(\boldsymbol{x})}{\partial x_n} \right) = \frac{\partial \phi(\boldsymbol{x})}{\partial \boldsymbol{x}^\top},$$

*the derivative (or Jacobian matrix) of $\boldsymbol{f}$ at $\boldsymbol{x}$ is the $m \times n$ matrix*

$$\mathsf{D}\boldsymbol{f}(\boldsymbol{x}) = \begin{pmatrix} \frac{\partial f_1(\boldsymbol{x})}{\partial x_1} & \cdots & \frac{\partial f_1(\boldsymbol{x})}{\partial x_n}, \\ \vdots & & \vdots \\ \frac{\partial f_m(\boldsymbol{x})}{\partial x_1} & \cdots & \frac{\partial f_m(\boldsymbol{x})}{\partial x_n} \end{pmatrix} = \frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}^\top},$$

*and the derivative (or Jacobian matrix) of $\mathbf{F}$ at $\mathbf{X}$ is the $mp \times nq$ matrix*

$$\mathsf{D}\mathbf{F}(\mathbf{X}) = \frac{\partial \mathrm{vec}\mathbf{F}(\mathbf{X})}{\partial (\mathrm{vec}\mathbf{X})^\top}.$$

We also use the following notation for the matrix derivatives of scalar function $\phi$ of two matrix arguments $\mathbf{X} = (x_{ij})$ and $\mathbf{Y} = (y_{ij})$

$$\frac{\mathrm{d}\phi(\mathbf{X}, \mathbf{Y})}{\mathrm{d}\mathbf{X}} \stackrel{\text{def}}{=} \mathsf{D}_{\mathbf{X}}\phi(\mathbf{X}, \mathbf{Y}) = \frac{\partial \phi(\mathbf{X}, \mathbf{Y})}{\partial (\mathrm{vec}\mathbf{X})^\top},$$
$$\frac{\mathrm{d}\phi(\mathbf{X}, \mathbf{Y})}{\mathrm{d}\mathbf{X}\mathrm{d}\mathbf{Y}} \stackrel{\text{def}}{=} \mathsf{D}_{\mathbf{X}} \left\{ \mathsf{D}_{\mathbf{Y}}\phi(\mathbf{X}, \mathbf{Y}) \right\}^\top,$$

where $\mathsf{D}_{\mathbf{X}}$ and $\mathsf{D}_{\mathbf{Y}}$ stress that the derivatives are with respect to $\mathbf{X}$ and $\mathbf{Y}$, respectively. The transpose sign of a row vector $\mathsf{D}_{\mathbf{Y}}\phi(\mathbf{X},\mathbf{Y})$ in the second formula is necessary since, in this framework, the calculus is developed for column vector valued functions.

Regarding the previous comment, in matrix calculus attention should be paid to the dimension of the matrix. Taking the derivative of the matrix is not the same as taking the derivative of the transpose matrix. Indeed, for the matrix $\mathbf{X}$ the derivative of the transpose function $\mathbf{F}(\mathbf{X}) = \mathbf{X}^\top$ is not an identity matrix, but it is given by $\mathsf{D}\mathbf{F}(\mathbf{X}) = \mathbf{K}_p$. For more on this subject see Magnus and Neudecker (2007), on which our discussion is based on and which also contains the following results that we use.

**Lemma 3.7.** *Let* $\mathbf{X}$ *be a square matrix of order* $p$, $\mathbf{A}$ *be a constant matrix of order* $p$ *and* $\mathbf{I}_{p^2}$ *and* $\mathbf{O}_{p^2}$ *the identity and the zero matrix of order* $p^2$*, respectively. The following identities hold:*

$$\mathsf{D}|\mathbf{X}| = |\mathbf{X}|\{\mathrm{vec}(\mathbf{X}^{-1})^\top\}^\top, \tag{3.24}$$

$$\mathsf{D}\mathrm{tr}(\mathbf{A}\mathbf{X}) = (\mathrm{vec}\mathbf{A}^\top)^\top, \tag{3.25}$$

$$\mathsf{D}\mathrm{vec}(\mathbf{X}) = \mathbf{I}_{p^2}, \tag{3.26}$$

$$\mathsf{D}\mathbf{X}^{-1} = -(\mathbf{X}^\top)^{-1} \otimes \mathbf{X}^{-1}, \tag{3.27}$$

$$\mathsf{D}\mathbf{A} = \mathbf{O}_{p^2}. \tag{3.28}$$

As a final remark, we note that a good reference on an alternative approach to matrix derivatives, which is frequently used, is given in (Turkington, 2005). Another useful reference is Harville (2008).

## 3.C Calculation of the derivatives

Recall that $l_k(\boldsymbol{\Theta}) = \frac{1}{2}\{\log|\boldsymbol{\Theta}| - \mathrm{tr}(\boldsymbol{\Theta}\mathbf{S}_k)\}$ and $f(\mathbf{S},\boldsymbol{\Theta}) = \log|\boldsymbol{\Theta}| - \mathrm{tr}(\mathbf{S}\boldsymbol{\Theta})$. Throughout the section we use the fact that the matrices $\boldsymbol{\Theta}$ and $\mathbf{S}_k$ are symmetric.

The chain rule and formulas (3.24) and (3.25) imply

$$\mathsf{D}l_k(\boldsymbol{\Theta}) = \frac{1}{2}\mathrm{vec}(\boldsymbol{\Theta}^{-1} - \mathbf{S}_k)^\top,$$

$$\frac{\mathrm{d}f(\mathbf{S},\boldsymbol{\Theta})}{d\boldsymbol{\Theta}} = \mathrm{vec}(\boldsymbol{\Theta}^{-1} - \mathbf{S})^\top.$$

Since $\boldsymbol{\psi}(\boldsymbol{y}_k, \boldsymbol{\Theta}) = \mathrm{vec}\mathsf{D}l_k(\boldsymbol{\Theta})$ it follows that

$$\boldsymbol{\psi}(\boldsymbol{y}_k, \boldsymbol{\Theta}) = \frac{1}{2}\mathrm{vec}(\boldsymbol{\Theta}^{-1} - \mathbf{S}_k).$$

The formulas (3.26), (3.27) and (3.28), and equality $(\mathbf{A} \otimes \mathbf{B})^\top = \mathbf{A}^\top \otimes \mathbf{B}^\top$ yield

$$\{\mathsf{D}\boldsymbol{\psi}(y_k; \boldsymbol{\Theta})\}^\top = -\frac{1}{2}\boldsymbol{\Theta}^{-1} \otimes \boldsymbol{\Theta}^{-1},$$
$$\frac{\mathrm{d}^2 f(\mathbf{S}, \boldsymbol{\Theta})}{\mathrm{d}\boldsymbol{\Theta}^2} = -\boldsymbol{\Theta}^{-1} \otimes \boldsymbol{\Theta}^{-1}.$$

Finally, $\mathrm{d}^2 f(\mathbf{S}, \boldsymbol{\Theta})/\mathrm{d}\boldsymbol{\Theta}\mathrm{d}\mathbf{S} = -\mathbf{I}_{p^2}$ follows from (3.26) and (3.28).

## 3.D Derivation of the expression for Q

We have that

$$\sum_{k=1}^{n} \boldsymbol{\psi}(\boldsymbol{y}_k, \widehat{\boldsymbol{\Theta}}) = \frac{1}{2}\sum_{k=1}^{n} \mathrm{vec}\Big(\widehat{\boldsymbol{\Theta}}^{-1} - \mathbf{S}_k\Big) = \mathbf{0}_{p^2}.$$

Applying the last equality and $\widehat{\boldsymbol{\Theta}}^{-1} = \mathbf{S}$ we obtain

$$\begin{aligned}
\mathbf{Q} &= \frac{1}{4n}\sum_{k=1}^{n} \mathrm{vec}\Big(\widehat{\boldsymbol{\Theta}}^{-1} - \mathbf{S}_k\Big)\mathrm{vec}(\widehat{\boldsymbol{\Theta}}^{-1} - \mathbf{S}_k)^\top \\
&= \frac{1}{4n}\left\{\sum_{k=1}^{n} \mathrm{vec}\left(\widehat{\boldsymbol{\Theta}}^{-1} - \mathbf{S}_k\right)\right\}\mathrm{vec}(\widehat{\boldsymbol{\Theta}}^{-1})^\top + \frac{1}{4n}\sum_{k=1}^{n} \mathrm{vec}\left(\widehat{\boldsymbol{\Theta}}^{-1} - \mathbf{S}_k\right)\mathrm{vec}(\mathbf{S}_k)^\top \\
&= \frac{1}{4n}\sum_{k=1}^{n} \mathrm{vec}\Big(\widehat{\boldsymbol{\Theta}}^{-1} - \mathbf{S}_k\Big)\mathrm{vec}(\mathbf{S}_k)^\top \\
&= \frac{1}{4n}\sum_{k=1}^{n} (\mathrm{vec}\mathbf{S} - \mathrm{vec}\mathbf{S}_k)\mathrm{vec}(\mathbf{S}_k)^\top \\
&= \frac{1}{4n}\mathrm{vec}\mathbf{S}\sum_{k=1}^{n} \mathrm{vec}(\mathbf{S}_k)^\top - \frac{1}{4n}\sum_{k=1}^{n} \mathrm{vec}\mathbf{S}_k\mathrm{vec}(\mathbf{S}_k)^\top \\
&= \frac{1}{4}\mathrm{vec}\mathbf{S}\mathrm{vec}(\mathbf{S})^\top - \frac{1}{4n}\sum_{k=1}^{n} \mathrm{vec}\mathbf{S}_k\mathrm{vec}(\mathbf{S}_k)^\top.
\end{aligned}$$

## 3.E Proof of Lemma 3.1

Substituting $\mathbf{M}_p = (\mathbf{I}_{p^2} + \mathbf{K}_p)/2$ in the equality (3.9) we obtain that it is equivalent to

$$(\boldsymbol{\Theta} \otimes \boldsymbol{\Theta})\mathrm{vec}\mathbf{A} = \mathbf{K}_p(\boldsymbol{\Theta} \otimes \boldsymbol{\Theta})\mathrm{vec}(\mathbf{A}).$$

To show this, we use identities $\text{vec}(\mathbf{ABC}) = (\mathbf{C}^\top \otimes \mathbf{A})\text{vec}\mathbf{B}$, $\mathbf{K}_p\text{vec}\mathbf{A} = \text{vec}\mathbf{A}^\top$ and symmetry of $\mathbf{A}$ and $\mathbf{\Theta}$

$$\mathbf{K}_p\mathbf{\Theta} \otimes \mathbf{\Theta}\text{vec}\mathbf{A} = \mathbf{K}_p\text{vec}(\mathbf{\Theta A \Theta}) = \text{vec}(\mathbf{\Theta A \Theta})^\top = \text{vec}(\mathbf{\Theta A \Theta}) = \mathbf{\Theta} \otimes \mathbf{\Theta}\text{vec}\mathbf{A}.$$

□

## 3.F Calculation of the algorithmic complexity

Throughout this section we use Lemma 4.5 from Section 4.C. We also use the result which follows from the definition of the Kronecker product.

**Lemma 3.8.** *If $\mathbf{\Theta}$ is a matrix of dimension $p \times p$ then the computational cost of $\mathbf{\Theta} \otimes \mathbf{\Theta}$ is $\mathcal{O}(p^4)$ flops.*

The matrices that appear in the calculations have the following dimensions:

- $\widehat{\mathbf{\Theta}}_\lambda, \mathbf{A}, \mathbf{B}, \mathbf{I}_\lambda$ are of dimension $p \times p$.
- $\widehat{\mathbf{\Theta}}_\lambda \otimes \widehat{\mathbf{\Theta}}_\lambda$ is of dimension $p^2 \times p^2$.
- $\text{vec}(\mathbf{A} \circ \mathbf{I}_\lambda)^\top$ is of dimension $1 \times p^2$.
- $\text{vec}(\mathbf{B} \circ \mathbf{I}_\lambda)$ and $\text{vec}\{\widehat{\mathbf{\Theta}}_\lambda(\mathbf{B} \circ \mathbf{I}_\lambda)\widehat{\mathbf{\Theta}}_\lambda\}$ are of dimension $p^2 \times 1$.

We do not take into account vectorization and transpose operator since they are connected with memory allocation and also because in 3.4 we show that these can be avoided.

**The cost for the term** $\text{vec}(\mathbf{A} \circ \mathbf{I}_\lambda)^\top(\widehat{\mathbf{\Theta}}_\lambda \otimes \widehat{\mathbf{\Theta}}_\lambda)\text{vec}(\mathbf{B} \circ \mathbf{I}_\lambda)$

We have that:

- $\widehat{\mathbf{\Theta}}_\lambda \mapsto \widehat{\mathbf{\Theta}}_\lambda \otimes \widehat{\mathbf{\Theta}}_\lambda$ costs $p^4$ flops.
- $(\mathbf{A}, \mathbf{I}_\lambda) \mapsto \text{vec}(\mathbf{A} \circ \mathbf{I}_\lambda)$ costs $p^2$ flops.
- $(\mathbf{B}, \mathbf{I}_\lambda) \mapsto \text{vec}(\mathbf{B} \circ \mathbf{I}_\lambda)$ costs $p^2$ flops.
- $(\widehat{\mathbf{\Theta}}_\lambda \otimes \widehat{\mathbf{\Theta}}_\lambda, \text{vec}(\mathbf{B} \circ \mathbf{I}_\lambda)) \mapsto (\widehat{\mathbf{\Theta}}_\lambda \otimes \widehat{\mathbf{\Theta}}_\lambda)\text{vec}(\mathbf{B} \circ \mathbf{I}_\lambda)$ costs $\mathcal{O}(p^4)$ flops.
- $(\text{vec}(\mathbf{A} \circ \mathbf{I}_\lambda)^\top, (\widehat{\mathbf{\Theta}}_\lambda \otimes \widehat{\mathbf{\Theta}}_\lambda)\text{vec}(\mathbf{B} \circ \mathbf{I}_\lambda)) \mapsto \text{vec}(\mathbf{A} \circ \mathbf{I}_\lambda)^\top(\widehat{\mathbf{\Theta}}_\lambda \otimes \widehat{\mathbf{\Theta}}_\lambda)\text{vec}(\mathbf{B} \circ \mathbf{I}_\lambda)$ costs $p^2$ flops.

Thus, the total computational cost is $\mathcal{O}(2p^4 + 3p^2)$, which is equal to $\mathcal{O}(p^4)$ flops.

**The cost for the term** $\text{vec}(\mathbf{A} \circ \mathbf{I}_\lambda)^\top\text{vec}\{\widehat{\mathbf{\Theta}}_\lambda(\mathbf{B} \circ \mathbf{I}_\lambda)\widehat{\mathbf{\Theta}}_\lambda\}$

It has already been shown that calculation of $\text{vec}(\mathbf{A} \circ \mathbf{I}_\lambda)$ and $\text{vec}(\mathbf{B} \circ \mathbf{I}_\lambda)$ costs a total of $2p^2$ flops. Furthermore, it holds:

- $(\mathbf{B} \circ \mathbf{I}_\lambda, \widehat{\mathbf{\Theta}}_\lambda) \mapsto (\mathbf{B} \circ \mathbf{I}_\lambda)\widehat{\mathbf{\Theta}}_\lambda$ costs $p^3$ flops.

- $(\widehat{\boldsymbol{\Theta}}_\lambda, (\mathbf{B} \circ \mathbf{I}_\lambda)\widehat{\boldsymbol{\Theta}}_\lambda) \mapsto \widehat{\boldsymbol{\Theta}}_\lambda(\mathbf{B} \circ \mathbf{I}_\lambda)\widehat{\boldsymbol{\Theta}}_\lambda$ costs $p^3$ flops.
- $(\text{vec}(\mathbf{A} \circ \mathbf{I}_\lambda)^\top, \text{vec}\{\widehat{\boldsymbol{\Theta}}_\lambda(\mathbf{B} \circ \mathbf{I}_\lambda)\widehat{\boldsymbol{\Theta}}_\lambda\}) \mapsto \text{vec}(\mathbf{A} \circ \mathbf{I}_\lambda)^\top \text{vec}\{\widehat{\boldsymbol{\Theta}}_\lambda(\mathbf{B} \circ \mathbf{I}_\lambda)\widehat{\boldsymbol{\Theta}}_\lambda\}$ costs $\mathcal{O}(p^2)$ flops.

Thus, the total computational cost is $\mathcal{O}(2p^3 + 3p^2)$, which is equal to $\mathcal{O}(p^3)$ flops.

# Chapter 4

# Time course window estimator for ordinary differential equations linear in the parameters

The subject of this chapter is the estimation of parameters in autonomous systems of ordinary differential equations. We consider systems that have a special structure, those for which the vector field is linear in the parameters. The structure of these systems allows construction of an estimator that has an explicit form. By using this estimator we not only avoid the usage of numerical ODEs solvers but optimization methods as well. As a result the estimator that we present is extremely fast. This is very important, since in Bayesian and likelihood approaches for estimating parameters of ODEs, the speed and convergence of the procedure may crucially depend on good initial values of the parameters, which this method can provide. Moreover, we prove that the proposed estimator is $\sqrt{n}$-consistent. The estimator does not require an initial guess for the parameters and is computationally fast, and therefore can serve as a good initial estimate for more efficient estimators. We illustrate our results in simulation studies .

The remainder of this chapter is divided into six sections. The problem is introduced in Section 4.1. In Section 4.2 we define the time course window estimator and present the formulae for the estimators of the parameters. In Section 4.3 we show results for different examples. Section 4.4 deals with the computational speed and complexity of the estimation procedure. In Section 4.5 we illustrate our method on a real data set. We conclude with a discussion in section 4.6. The last section contains the summary. All proofs and auxiliary results are given in the Appendix.

## 4.1 Introduction

Consider the system of differential equations of the form

$$\begin{cases} \boldsymbol{x}'(t) = \boldsymbol{f}(\boldsymbol{x}(t); \boldsymbol{\theta}), \quad t \in [0, T], \\ \boldsymbol{x}(0) = \boldsymbol{\xi}, \end{cases} \tag{4.1}$$

where $\boldsymbol{x}(t)$ takes values in $\mathbb{R}^d$, $\boldsymbol{\xi}$ in $\Xi \subset \mathbb{R}^d$, and $\boldsymbol{\theta}$ in $\Theta \subset \mathbb{R}^p$. From (4.1) we obtain the system of integral equations

$$\boldsymbol{x}(t) = \boldsymbol{\xi} + \int_0^t \boldsymbol{f}(\boldsymbol{x}(s); \boldsymbol{\theta}) \, \mathrm{d}s, \, t \in [0, T]. \tag{4.2}$$

Given the values of $\boldsymbol{\xi}$ and $\boldsymbol{\theta}$, we denote the solution of (4.1)-(4.2) by $\boldsymbol{x}(t; \boldsymbol{\theta}, \boldsymbol{\xi})$. In many practical applications, the values of $\boldsymbol{\xi}$ and $\boldsymbol{\theta}$ are unknown and need to be estimated from the data, which consist of noisy observations of $\boldsymbol{x}(t; \boldsymbol{\theta}, \boldsymbol{\xi})$ at certain time points in $[0, T]$. We denote the observations by

$$\boldsymbol{y}(t_i) = \boldsymbol{x}(t_i; \boldsymbol{\theta}, \boldsymbol{\xi}) + \boldsymbol{\varepsilon}(t_i), \quad i = 1, \ldots, n, \tag{4.3}$$

where for simplicity $t_i = iT/n, \; i = 1, \ldots, n$ and $\boldsymbol{\varepsilon}(t_i)$, are the $d$-dimensional column vectors of measurement errors at time $t_i$. We focus on the special class of nonlinear systems that are linear in the parameter $\boldsymbol{\theta}$

$$\boldsymbol{f}(\boldsymbol{x}(t); \boldsymbol{\theta}) = \mathbf{g}(\boldsymbol{x}(t))\boldsymbol{\theta}, \tag{4.4}$$

where the measurable function $\mathbf{g} : \mathbb{R}^d \to \mathbb{R}^{d \times p}$ maps the $d$-dimensional column vector $\boldsymbol{x}$ into a $d \times p$ matrix. In this chapter, we consider estimators of the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\xi}$ that are obtained by minimizing

$$L_n(\boldsymbol{\xi}, \boldsymbol{\theta}) = \int_0^T \left\| \widehat{\boldsymbol{x}}_n(t) - \boldsymbol{\xi} - \int_0^t \mathbf{g}(\widehat{\boldsymbol{x}}_n(s)) \, \mathrm{d}s \, \boldsymbol{\theta} \right\|^2 \mathrm{d}t, \tag{4.5}$$

with respect to $\boldsymbol{\xi}$ and $\boldsymbol{\theta}$, where $\widehat{\boldsymbol{x}}_n(t), t \in [0, T]$, is a particular estimator of $\boldsymbol{x}(t; \boldsymbol{\theta}, \boldsymbol{\xi})$. Here, $\| \cdot \|$ denotes the Euclidean norm $\|\boldsymbol{z}\| = \{\sum_{j=1}^d z_j^2\}^{1/2}$. Minimization of (4.5) results in an explicit form for the estimators $\widehat{\boldsymbol{\theta}}_n$ and $\widehat{\boldsymbol{\xi}}_n$, which is not the case with classical approaches, such as non-linear least squares (NLS) or maximum likelihood estimator (MLE). Indeed, NLS and MLE require the solution of the system of differential equations, which generally does not have a closed form. This remains true even

when the dependence on the model parameters is linear.

In case of repeated measures, using a step function estimator for $\widehat{\boldsymbol{x}}_n(t)$ yields a very simple and computationally fast estimator. The estimators of the parameters are $\sqrt{n}$-consistent if, roughly speaking, the number of time points is of order $\sqrt{n}$ and for most time points the number of replicates is of the same order (Dattner and Klaassen, 2013). In case of time course data without repeated measurements, as introduced above, the aforementioned consistency result does not hold unless some smoothing is applied. In this chapter we examine this case. We show that by defining a suitable step-function estimator of the solution of the system $\boldsymbol{x}$, we obtain explicit estimators of the parameters that are $\sqrt{n}$-consistent. This parametric convergence rate is obtained by using only weak assumptions on the measurement error.

The idea of smoothing as a way to avoid numerical integration of the system of differential equations has been used before and is referred to as the *collocation* estimation method; for example, there are *two-step* methods (Bellman and Roth, 1971; Varah, 1982; Brunel, 2008; Liang and Wu, 2008; Fang et al., 2011; Gugushvili and Klaassen, 2012; Gugushvili and Spreij, 2012; Dattner and Klaassen, 2013) and *generalized profiling* methods (Ramsay et al., 2007; Qi and Zhao, 2010; Hooker et al., 2011; Xun et al., 2013). Also, other regularization based approaches, which make use of properties of differential operators, have been proposed to avoid numerical integration of the system of differential equations (Steinke and Schölkopf, 2008). In most cases, the main computational bottleneck lies in the optimization of a non-linear objective function. Indeed, standard problems connected with optimization also appear here. For example, a poor initial guess may lead to considerably slower convergence to the global optimum or even to some local optimum. For systems that are linear in the parameters this optimization can be avoided, since the estimator can be obtained explicitly (Himmelblau et al., 1967; Dattner and Klaassen, 2013). Moreover, following Khanin et al. (2006, 2007) further simplifications can be obtained by assuming piecewise constant solutions of the differential equations.

## 4.2 Time-course window estimator

Let $t_1, \ldots, t_n$ be equidistant time points of the observations. We divide the interval $[0, T]$ into $I = \lfloor \sqrt{n} \rfloor$ subintervals of the length $\Delta = T/I$ so that in every interval we have at least $\lfloor \sqrt{n} \rfloor$ and at most $\lfloor \sqrt{n} \rfloor + 2$ time points, i.e. $I$, $I+1$ or $I+2$ points. Let $S_i = [a_{i-1}, a_i)$ be the $i$th subinterval $i = 1, \ldots, I-1$ and $S_I = [a_{I-1}, a_I]$. The boundary

points of the subintervals are $a_i = i\Delta$, $i = 0, \dots, I$ and endpoints of $[0, T]$ are $a_0 = 0$ and $a_I = T$. For $t \in [0, T]$, let $S(t)$ denote the subinterval to which $t$ belongs. In other words, if $t \in S_i$ then $S(t) = S_i$. The window estimator of $\boldsymbol{x}$ is defined as

$$\widehat{\boldsymbol{x}}_n(t) = \frac{1}{|S(t)|} \sum_{t_j \in S(t)} \boldsymbol{y}(t_j), \qquad t \in S(t). \tag{4.6}$$

This estimator is a stepwise function that estimates $\boldsymbol{x}(t)$ in each interval as the mean of the observations that belong to that interval. On each interval $S_i$, the window estimator $\widehat{\boldsymbol{x}}_n$ takes a constant value that we denote by $\widehat{\boldsymbol{x}}_n(S_i)$. In other words it holds

$$\widehat{\boldsymbol{x}}_n(t) = \widehat{\boldsymbol{x}}_n(S_i), \qquad t \in S_i. \tag{4.7}$$

We estimate $\mathbf{G}(t) = \int_0^t \mathbf{g}(\boldsymbol{x}(s))\mathrm{d}s$ via a plug-in approach

$$\widehat{\mathbf{G}}_n(t) = \int_0^t \mathbf{g}(\widehat{\boldsymbol{x}}_n(s))\mathrm{d}s.$$

This is just a finite sum as it is the integral of a piecewise constant function. Indeed, since $\widehat{\boldsymbol{x}}_n(\cdot)$ is piecewise constant therefore $\mathbf{g}(\widehat{\boldsymbol{x}}_n(\cdot))$ is as well. This gives

$$\widehat{\mathbf{G}}_n(t) = \sum_{m=1}^{i-1} \mathbf{g}(\widehat{\boldsymbol{x}}_n(S_m))\Delta + \mathbf{g}(\widehat{\boldsymbol{x}}_n(S_i))(t - a_{i-1}), \qquad t \in S_i.$$

Throughout the chapter, we adhere to the convention that the sums of the form $\sum_{m=1}^{i-1} f_m$ are equal to zero for $i = 1$. Minimizing the criterion function (4.5) with respect to $\boldsymbol{\xi}$ and $\boldsymbol{\theta}$ yields explicit formulae for the estimators of the parameters. Indeed, the objective function $L_n$ can be written as

$$L_n(\boldsymbol{\omega}) = \boldsymbol{\omega}^\top \int_0^T \mathbf{F}(t)^\top \mathbf{F}(t)\mathrm{d}t\boldsymbol{\omega} - 2\boldsymbol{\omega}^\top \int_0^T \mathbf{F}(t)^\top \widehat{\boldsymbol{x}}_n(t)\mathrm{d}t + \int_0^T \|\widehat{\boldsymbol{x}}_n(t)\|^2 \mathrm{d}t,$$

where $\boldsymbol{\omega} = (\boldsymbol{\xi}; \boldsymbol{\theta})^\top$ and $\mathbf{F}(t) = (T\mathbf{I}_d; \widehat{\mathbf{G}}_n(t))$. Thus, $L_n$ is quadratic in $\boldsymbol{\omega}$ and its minimizer is

$$\widehat{\boldsymbol{\omega}} = \left( \int_0^T \mathbf{F}(t)^\top \mathbf{F}(t)\mathrm{d}t \right)^{-1} \int_0^T \mathbf{F}(t)^\top \widehat{\boldsymbol{x}}_n(t)\mathrm{d}t = \begin{pmatrix} T\mathbf{I}_d & \widehat{\mathbf{A}}_n \\ \widehat{\mathbf{A}}_n^\top & \widehat{\mathbf{B}}_n \end{pmatrix}^{-1} \begin{pmatrix} \int_0^T \widehat{\boldsymbol{x}}_n(t)\mathrm{d}t \\ \int_0^T \widehat{\mathbf{G}}_n(t)^\top \widehat{\boldsymbol{x}}_n(t)\mathrm{d}t \end{pmatrix},$$

where

$$\widehat{\mathbf{A}}_n = \int_0^T \widehat{\mathbf{G}}_n(t)\mathrm{d}t,$$
$$\widehat{\mathbf{B}}_n = \int_0^T \widehat{\mathbf{G}}_n(t)^\top \widehat{\mathbf{G}}_n(t)\mathrm{d}t.$$

By using the matrix block inversion (Bernstein, 2009, Chapter 2) we obtain

$$\begin{aligned}\widehat{\boldsymbol{\xi}}_n &= \left(T\mathbf{I}_d - \widehat{\mathbf{A}}_n\widehat{\mathbf{B}}_n^{-1}\widehat{\mathbf{A}}_n^\top\right)^{-1}\int_0^T \left\{\mathbf{I}_d - \widehat{\mathbf{A}}_n\widehat{\mathbf{B}}_n^{-1}\widehat{\mathbf{G}}_n(t)^\top\right\}\widehat{\boldsymbol{x}}_n(t)\mathrm{d}t,\\ \widehat{\boldsymbol{\theta}}_n &= \widehat{\mathbf{B}}_n^{-1}\int_0^T \widehat{\mathbf{G}}_n(t)^\top\{\widehat{\boldsymbol{x}}_n(t) - \widehat{\boldsymbol{\xi}}_n\}\mathrm{d}t.\end{aligned} \tag{4.8}$$

Plugging (4.7) into (4.8) we obtain

$$\begin{aligned}\widehat{\boldsymbol{\xi}}_n &= \left(T\mathbf{I}_d - \widehat{\mathbf{A}}_n\widehat{\mathbf{B}}_n^{-1}\widehat{\mathbf{A}}_n^\top\right)^{-1}\left\{T\textstyle\sum_{i=1}^I \widehat{\boldsymbol{x}}_n(S_i)/I - \widehat{\mathbf{A}}_n\widehat{\mathbf{B}}_n^{-1}\sum_{i=1}^I \widehat{\mathbf{A}}_n(i)^\top\widehat{\boldsymbol{x}}_n(S_i)\right\},\\ \widehat{\boldsymbol{\theta}}_n &= \widehat{\mathbf{B}}_n^{-1}\left\{\textstyle\sum_{i=1}^I \widehat{\mathbf{A}}_n(i)^\top\widehat{\boldsymbol{x}}_n(S_i) - \widehat{\mathbf{A}}_n^\top\widehat{\boldsymbol{\xi}}_n\right\},\end{aligned}$$

where $\widehat{\mathbf{A}}_n(i) = \int_{S_i}\widehat{\mathbf{G}}_n(t)\mathrm{d}t$. The integrals $\widehat{\mathbf{A}}_n$, $\widehat{\mathbf{B}}_n$, $\widehat{\mathbf{A}}_n(i)$ have explicit forms in terms of $\widehat{\mathbf{G}}_n(a_i)$ and $\mathbf{g}(\widehat{\boldsymbol{x}}_n(S_i))$:

$$\widehat{\mathbf{A}}_n = T\widehat{\mathbf{G}}_n(T) - \frac{\Delta^2}{2}\sum_{i=1}^I(2i-1)\mathbf{g}(\widehat{\boldsymbol{x}}_n(S_i)),$$

$$\widehat{\mathbf{A}}_n(i) = t\widehat{\mathbf{G}}_n(t)|_{a_{i-1}}^{a_i} - \frac{1}{2}(2i-1)\Delta^2\mathbf{g}(\widehat{\boldsymbol{x}}_n(S_i)), \tag{4.9}$$

and

$$\begin{aligned}\widehat{\mathbf{B}}_n &= \widehat{\mathbf{G}}_n(T)^\top\widehat{\mathbf{A}}_n - \frac{1}{2}\sum_{i=1}^I \mathbf{g}(\widehat{\boldsymbol{x}}_n(S_i))^\top t^2\widehat{\mathbf{G}}_n(t)\Big|_{a_{i-1}}^{a_i}\\ &+ \frac{\Delta^3}{6}\sum_{i=1}^I(3i^2-1)\mathbf{g}(\widehat{\boldsymbol{x}}_n(S_i))^\top\mathbf{g}(\widehat{\boldsymbol{x}}_n(S_i))\\ &+ \frac{\Delta^3}{2}\sum_{i=1}^{I-1}(2i-1)\mathbf{R}_i^\top\mathbf{g}(\widehat{\boldsymbol{x}}_n(S_i)),\end{aligned} \tag{4.10}$$

where $\mathbf{R}_i = \sum_{m=i+1}^I \mathbf{g}(\widehat{\boldsymbol{x}}_n(S_m))$. See appendix 4.D for the derivations of these formulas.

Next, we formulate the $\sqrt{n}$-consistency of $\widehat{\boldsymbol{\xi}}_n$ and $\widehat{\boldsymbol{\theta}}_n$ in the following theorem. A prerequisite for consistent estimation is the identifiability of the parameters. There are several concepts of identifiability. Here we are concerned with *structural identifiability*, a property that depends on the structure of the model and is not affected by the

experimental set-up. This means that knowledge of the solution $\{\boldsymbol{x}(t),\, t \in [0,T]\}$ yields the unique values of the parameters $\boldsymbol{\xi}$ and $\boldsymbol{\theta}$. For $\boldsymbol{\xi} = \boldsymbol{x}(0)$ this is true, while identifiability for $\boldsymbol{\theta}$ means that $\boldsymbol{\theta}' \neq \boldsymbol{\theta} \Rightarrow \boldsymbol{x}(\cdot;\boldsymbol{\theta}',\boldsymbol{\xi}) \neq \boldsymbol{x}(\cdot;\boldsymbol{\theta},\boldsymbol{\xi})$; see Dattner and Klaassen (2013) for a necessary and sufficient condition for identifiability in the case of systems linear in the parameters.

**Theorem 4.1.** *Let an ODE model be defined by (4.1), (4.2) and (4.4) with the twice continuously differentiable map* $\mathbf{g} : \mathbb{R}^d \to \mathbb{R}^d \times \mathbb{R}^p$. *Let* $\boldsymbol{\xi} \in \Xi$ *and* $\boldsymbol{\theta} \in \Theta$ *and let* $\boldsymbol{x}(\cdot) = \boldsymbol{x}(\cdot;\boldsymbol{\theta},\boldsymbol{\xi})$ *exist and be bounded on* $[0,T]$. *Assume that* $\boldsymbol{\theta}$ *is identifiable. Let the observations be given by (4.3) with* $t_i = iT/n,\ i = 1,...,n$. *Assume that* $\varepsilon_k(t_i),\ k = 1,\dots,d,\ i = 1,\dots,n$, *are i.i.d. random variables with zero expectation and finite variance* $\sigma^2$. *Let* $\widehat{\boldsymbol{x}}_n(\cdot)$ *be given by (4.6) and let* $\widehat{\boldsymbol{\theta}}_n, \widehat{\boldsymbol{\xi}}_n$ *be defined as in (4.8). Then*

$$(\widehat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}, \widehat{\boldsymbol{\xi}}_n - \boldsymbol{\xi}) = O_p(n^{-1/2}), \quad n \to \infty,$$

*holds.*

Below we sketch the proof of this theorem. The full proof is deferred to the appendix. The main idea is that if $\widehat{\boldsymbol{x}}_n(\cdot)$ is a consistent estimator of $\boldsymbol{x}(\cdot)$ in the sup norm, then the estimators $\widehat{\boldsymbol{\theta}}_n$ and $\widehat{\boldsymbol{\xi}}_n$ are also consistent. If, in fact, the estimator $\widehat{\boldsymbol{x}}_n(\cdot)$ satisfies some stronger conditions, then it even implies $\sqrt{n}$-consistency of $\widehat{\boldsymbol{\theta}}_n$ and $\widehat{\boldsymbol{\xi}}_n$.

The proof can be divided into two parts. In the first part we analyze the behaviour of the expected value of the window estimator and in the second we analyze its variance. In order to obtain the consistency of $\widehat{\boldsymbol{x}}_n$ we first need to show that the sup norm of $\widehat{\boldsymbol{x}}_n$ is bounded in probability, i.e.

$$\|\widehat{\boldsymbol{x}}_n\|_\infty = O_p(1),$$

and that the expected value of $\widehat{\boldsymbol{x}}_n$ converges to $\boldsymbol{x}(\cdot)$ in the sup norm. If $\mathbf{g}(\cdot)$ is bounded, then this condition is not needed. However, to obtain $\sqrt{n}$-consistency, convergence needs to be at least at $\sqrt{n}$ rate, i.e.

$$\|\mathrm{E}\widehat{\boldsymbol{x}}_n - \boldsymbol{x}\|_\infty = O(c_n),$$

where $c_n = n^{-1/2}$. The previous two equalities can be proven by using the boundedness of the parameter space, the linearity of the system in the parameters and Chebyshev's

inequality. Furthermore, we can show that

$$\int_0^T \|\widehat{\boldsymbol{x}}_n(t) - \mathrm{E}\widehat{\boldsymbol{x}}_n(t)\|^2 \mathrm{d}t = O_p(d_n),$$

where the convergence is again at rate $d_n = n^{-1/2}$. This can be shown by using the independence of the observations and the specific form of the window estimator.

To prove the conditions related to the variance of the window estimator, let $\phi : [0, T] \to \mathbb{R}$ be any bounded measurable function and denote the $k$th component of $\widehat{\boldsymbol{x}}_n(\cdot)$ by $\widehat{x}_{n,k}(\cdot)$, for $k = 1, \ldots, d$. Using the independence of the observations and the fact that the estimator is piecewise constant we can show that

$$\int_0^T \mathrm{var}\left\{\int_0^t \phi(s)\widehat{x}_{n,k}(s)\mathrm{d}s\right\}\mathrm{d}t = O\left(v_n^2\right),$$

$$\mathrm{var}\left\{\int_0^T \phi(t)\widehat{x}_{n,k}(t)\mathrm{d}t\right\} = O\left(v_n^2\right),$$

where $v_n = n^{-1/2}$. These conditions ensure that our estimator satisfies the "plug-in" properties studied in Goldstein and Messer (1992) and Bickel and Ritov (2003). By Theorem 2 (Dattner and Klaassen, 2013) it follows that

$$(\widehat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}, \widehat{\boldsymbol{\xi}}_n - \boldsymbol{\xi}) = O_p\left(c_n + d_n + v_n\right) = O_p(n^{-1/2}), n \to \infty.$$

## 4.3 Simulation examples

In this section we illustrate the performance of the estimators $\widehat{\boldsymbol{\xi}}_n$ and $\widehat{\boldsymbol{\theta}}_n$ and indicate how the method can be useful in combination with other methods. First, we show empirically the $\sqrt{n}$-consistency of the estimates $\widehat{\boldsymbol{\theta}}_n$ and $\widehat{\boldsymbol{\xi}}_n$ when the sample size increases. Secondly, we demonstrate the robustness of the estimator with respect to different measurement errors. Also, although primitive, the window-based estimator provides reasonably good estimates. Finally, we show how the window-based estimator can provide useful starting values for other methods, which typically require a sensible initial value.

| Gaussian error, $\sigma_1 = 43$, $\sigma_2 = 12.5$, $\sigma_3 = 2446$ | | | | |
|---|---|---|---|---|
| | Value | $n = 100$ | $n = 3000$ | $n = 10000$ |
| $\theta_1$ | 36.000 | 35.467 (7.602) | 35.987 (1.538) | 36.025 (0.841) |
| $\theta_2$ | 9.42e-06 | 9.42e-06 (0.182e-06) | 9.50e-06(0.032e-06) | 9.50e-06 ( 0.017e-06) |
| $\theta_3$ | 1000.000 | 958.006 (41.728) | 1000.358 (7.562) | 999.535 (4.107) |
| $\theta_4$ | 3.000 | 2.928 (0.058) | 3.000 (0.010) | 2.999 (0.006) |

Table 4.1 The empirical mean and standard deviation (in parentheses) of the window estimator of the parameters of the HIV dynamics model with Gaussian error. Results are based on 500 simulations.

### 4.3.1 Empirical validation of $\sqrt{n}$-consistency

In this subsection we consider a model for HIV viral fitness (Bonhoeffer et al., 1997). The basic model considers three types of agents: the uninfected cells $x_1$, the infected cells $x_2$ and the virus particles $x_3$. Uninfected cells are produced at constant rate $\theta_1$, and die at rate $0.108x_1$. Free virus infects uninfected cells to produce infected cells at rate $\theta_2 x_1 x_3$. Infected cells die at rate $0.5x_2$. New virus is produced from infected cells at rate $0.5\theta_3 x_2$ and dies at rate $\theta_4 x_3$. These assumptions lead to the following set of differential equations:

$$\begin{aligned} x_1'(t) &= \theta_1 - 0.108x_1(t) - \theta_2 x_1(t)x_3(t), \\ x_2'(t) &= \theta_2 x_1(t)x_3(t) - 0.5x_2(t), \\ x_3'(t) &= 0.5\theta_3 x_2(t) - \theta_4 x_3(t). \end{aligned}$$

Xue et al. (2010) consider the case where $x_1$ and $x_2$ cannot be measured separately while Miao et al. (2008) study a similar model in which all states are measured. Here we assume that all concentrations are measured, and following Xue et al. (2010) our goal is to estimate the parameter vector $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3, \theta_4)^\top$. The window-based method is applied for different sample sizes $n$, to data as in (4.3). Each simulation is repeated 500 times and the results shown are empirical means and standard deviations across those runs.

We first generate the solutions of $x_1, x_2, x_3$ over the time interval $[0, 1]$ based on $\xi_1 = 600$, $\xi_2 = 30$, $\xi_3 = 10^5$ and $\theta_1 = 36$, $\theta_2 = 9.5 \cdot 10^{-6}$, $\theta_3 = 1000$, $\theta_4 = 3$. We add noise to each solution with $\sigma_1 = 43$, $\sigma_2 = 12.5$, $\sigma_3 = 2446$. These noise levels correspond to approximately 20% of the average value (over the time interval) of $x_1$,

$x_2$ and $x_3$ respectively. Sample sizes are varied according to $n = 100, 3000, 10000$. The initial values $\boldsymbol{\xi}$ are considered known. The results are presented in Table 4.1. As Theorem 1 suggests, we see that as the sample size increases the accuracy of the estimates improves. The standard deviation of the estimators seems to follow a $1/\sqrt{n}$ behaviour. For example, scaling the empirical standard deviations of the parameter $\theta_3$ that correspond to sample sizes $n = 100, 3000, 10000$ with $\sqrt{n}$ yields values 417.280 414.188 and 410.700, respectively.

We also compare the results with those obtained in Xue et al. (2010), in terms of the average relative estimation error (ARE), which is defined as

$$\text{ARE} = \frac{1}{M}\sum_{j=1}^{M}\frac{|\widehat{\theta}_k^{(j)} - \theta_k|}{|\theta_k|} \times 100\%,$$

where $\widehat{\theta}_k^{(j)}$ is the estimate of the parameter $\theta_k$, $k = 1, \ldots, 4$, from the $j$th simulation data set and $M$ is the total number of simulation runs. Since the results in Xue et al. (2010) are obtained with sample size $n = 40$ over the time interval $[0, 19.9]$ we also did a simulation with 500 random data sets for the same sample size and time interval. We found the ARE for the parameters $\theta_1$, $\theta_3$, $\theta_4$ to be equal to 3.86, 54.98, 53.02 percent while the results in Xue et al. (2010) are 3.19, 17.7, 17.4 percent (ARE for $\theta_2$ is not given in Xue et al. (2010)).

### 4.3.2 Comparing different error distributions

Theorem 4.1 asserts that the $\sqrt{n}$-consistency holds for any error distribution with finite variance. Simulations presented in this section suggest that the convergence depends merely on the variance of the error distribution. In this section we compare the behaviour of the window-based estimator under Gaussian and Laplace noise. We consider the Lotka-Volterra system of differential equations, which is frequently used to describe the dynamics of biological systems in which two species, predators and their preys, interact (see Edelstein-Keshet (2005) for details on this system). The Lotka-Volterra system consists of two equations depending on the parameter vector $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3, \theta_4)^\top$. The system has the form

$$\begin{aligned} x_1'(t) &= \theta_1 x_1(t) - \theta_2 x_1(t) x_2(t), \\ x_2'(t) &= -\theta_3 x_2(t) + \theta_4 x_1(t) x_2(t). \end{aligned} \tag{4.11}$$

We select the parameter $(\theta_1, \theta_2, \theta_3, \theta_4)^\top = (0.1, 0.9, 0.9, 0.5)^\top$, the initial condition $(\xi_1, \xi_2)^\top = (1, 0.5)^\top$ and consider three sample size scenarios $n = 100, 3000, 10000$. The data sets are generated with i.i.d. Gaussian and Laplace noise with zero mean and standard deviation equal to 0.5 for both states. The results for both measurements errors and different sample sizes are shown in Table 4.2. The empirical means are similar in the case of both error distributions. In Figure 4.1, the window smoothers of $x_1$ and $x_2$ are presented for one realization of 100 equidistant observations on the interval [0, 49.9] in the case of Gaussian noise. As can been seen from the figure, the window smoother is rather primitive, a fact which explains why the estimator is inferior in performance in small sample sizes. Still, it can be used as an initial value for more efficient procedures. We illustrate this in the next section.

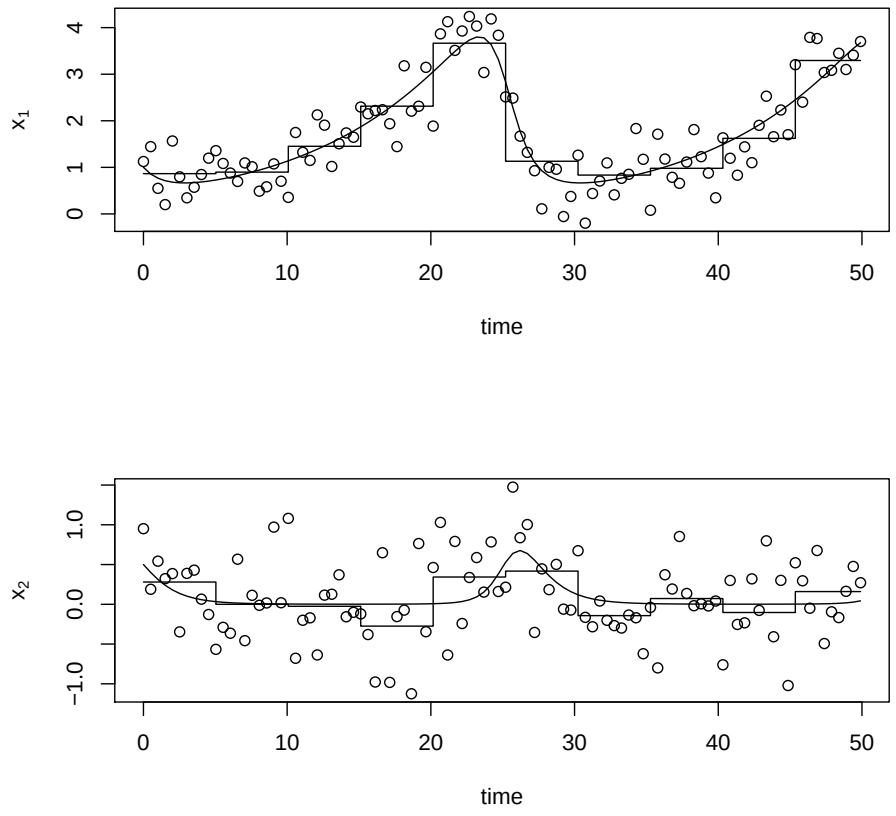



Fig. 4.1 The Lotka-Volterra system. The solid lines correspond to the states $x_1$ and $x_2$ as given by the model (4.11) with $\theta_1 = 0.1$, $\theta_2 = 0.9$, $\theta_3 = 0.9$, $\theta_4 = 0.5$. The bold step functions correspond to the window smoothers of $x_1$ and $x_2$. The data, represented by the circles in the figure, consist of 100 equidistant observations on the interval $[0, 49.9]$.

| Gaussian error, $S.D. = 0.50$ | | | | | | | |
|---|---|---|---|---|---|---|---|
| | Value | $n = 20$ | $n = 30$ | $n = 50$ | $n = 100$ | $n = 3000$ | $n = 10000$ |
| $\xi_1$ | 1.0 | 1.162 (0.339) | 1.080 (0.298) | 1.083 (0.329) | 1.146 (0.408) | 1.007 (0.145) | 0.999 (0.076) |
| $\xi_2$ | 0.5 | 0.146 (0.271) | 0.193 (0.281) | 0.168 (0.223) | 0.213 (0.206) | 0.478 (0.101) | 0.489 (0.055) |
| $\theta_1$ | 0.1 | 0.022 (0.040) | 0.031 (0.033) | 0.041 (0.031) | 0.053 (0.038) | 0.097 (0.010) | 0.099 (0.006) |
| $\theta_2$ | 0.9 | 0.013 (0.273) | 0.113 (0.204) | 0.237 (0.204) | 0.432 (0.258) | 0.877 (0.074) | 0.895 (0.045) |
| $\theta_3$ | 0.9 | 0.078 (0.231) | 0.115 (0.266) | 0.139 (0.215) | 0.230 (0.208) | 0.849 (0.109) | 0.881 (0.067) |
| $\theta_4$ | 0.5 | 0.047 (0.132) | 0.062 (0.130) | 0.071 (0.108) | 0.132 (0.139) | 0.472 (0.066) | 0.491 (0.040) |
| Laplace error, $S.D. = 0.50$ | | | | | | | |
| | Value | $n = 20$ | $n = 30$ | $n = 50$ | $n = 100$ | $n = 3000$ | $n = 10000$ |
| $\xi_1$ | 1.000 | 1.188 (0.311) | 1.085 (0.309) | 1.073 (0.343) | 1.118 (0.374) | 1.000 (0.140) | 0.999 (0.078) |
| $\xi_2$ | 0.500 | 0.147 (0.263) | 0.187 (0.284) | 0.188 (0.231) | 0.211 (0.217) | 0.474 (0.096) | 0.494 (0.055) |
| $\theta_1$ | 0.100 | 0.017 (0.032) | 0.033 (0.030) | 0.045 (0.033) | 0.053 (0.037) | 0.097 (0.011) | 0.099 (0.006) |
| $\theta_2$ | 0.900 | 0.021 (0.238) | 0.137 (0.191) | 0.257 (0.213) | 0.434 (0.289) | 0.878 (0.075) | 0.891 (0.044) |
| $\theta_3$ | 0.900 | 0.072 (0.216) | 0.130 (0.280) | 0.143 (0.224) | 0.238 (0.221) | 0.846 (0.104) | 0.883 (0.065) |
| $\theta_4$ | 0.500 | 0.043 (0.137) | 0.064 (0.138) | 0.073 (0.115) | 0.135 (0.140) | 0.472 (0.064) | 0.490 (0.040) |

Table 4.2 Empirical mean and standard deviation (in parentheses) of window estimator of parameters and initial values of Lotka-Volterra system with Gaussian and Laplace error. The results are based on 500 simulations.

### 4.3.3 Window-based estimator as initial estimate

To conclude the simulation section, we illustrate how the proposed method can be used to provide an initial value of the parameters for a more sophisticated procedure. In particular, here we use the window estimator as an initial value for *the generalized profiling* procedure (Ramsay et al., 2007), which is implemented in the R package CollocInfer (Hooker et al., 2012). We consider the *FhNdata* from this package, which is generated from the FitzHugh-Nagumo system given by the equations

$$\begin{aligned} x_1'(t) &= c\{x_1(t) - x_1(t)^3/3 + x_2(t)\}, \\ x_2'(t) &= -\tfrac{1}{c}\{x_1(t) - a + bx_2(t)\}. \end{aligned} \tag{4.12}$$

This system is used to model electrical activity in a neuron (FitzHugh, 1961; Nagumo et al., 1962). *FhNdata* consist of 41 equally spaced observations on the interval [0,20] obtained from the model (4.12) with noise levels $\sigma_1 = \sigma_2 = 0.5$ and parameters $a = 0.2$, $b = 0.2$ and $c = 3$. First, we run the generalized profiling procedure with different initial guesses. The procedure is tuned according to the CollocInfer manual. More specifically, we smooth the data with a roughness penalty approach by using B-splines of order 3. The knots of the spline are equally spaced with step size equal to 0.2, and the number of basis functions is one more than the number of knots. The B-spline tuning parameter is set to 0.5. Smoothing the data yields the initial values for the B-spline basis coefficients, which can be supplied to the generalized profiling method. Then the procedure is applied with a tuning parameter $\lambda = 10^4$ and initial guesses $10^{i-1}u$, $i = 0, \ldots, 5$, where $u = (1, 1, 1)^\top$. The FitzHugh-Nagumo system is not linear in the parameters so we cannot directly use the window estimator. However, following Dattner and Klaassen (2013), we define

$$\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3, \theta_4)^\top = (c, 1/c, a/c, b/c)^\top,$$

making the system linear in $\boldsymbol{\theta}$. The obtained window estimates are $\widehat{\theta}_1 = 0.160$, $\widehat{\theta}_2 = 0.333$, $\widehat{\theta}_3 = 0.106$ and $\widehat{\theta}_4 = -0.047$. Therefore, we obtain two estimates of $c$, namely, $\widehat{\theta}_1 = 0.160$ and $1/\widehat{\theta}_2 = 3.003$. This is not surprising since in the same experimental setup, but with 400 data points, Campbell and Steele (2012) showed, using MCMC techniques, that the posterior for $c$ has three modes around 0.5, 3 and 9. The estimates of $c$ which we have obtained are close to the first and second modes, respectively. We numerically solve the system of ODEs for these two estimates. As an initial condition we use its window estimate, which is $(\widehat{\xi}_1, \widehat{\xi}_2)^\top = (0.051, 0.569)^\top$. Visual

inspection of obtained solutions shows that the second parameter agrees better with the data (see Figure 4.2). Thus, for the window estimate we choose $(\widehat{a}_2, \widehat{b}_2, \widehat{c}_2)^\top = (0.318, -0.140, 3.003)^\top$. The results for the generalized profiling procedure for seven different initial guesses are presented in Table 4.3, showing that the window estimator can be a very useful method for providing a stable initial value to the generalized profiling procedure.

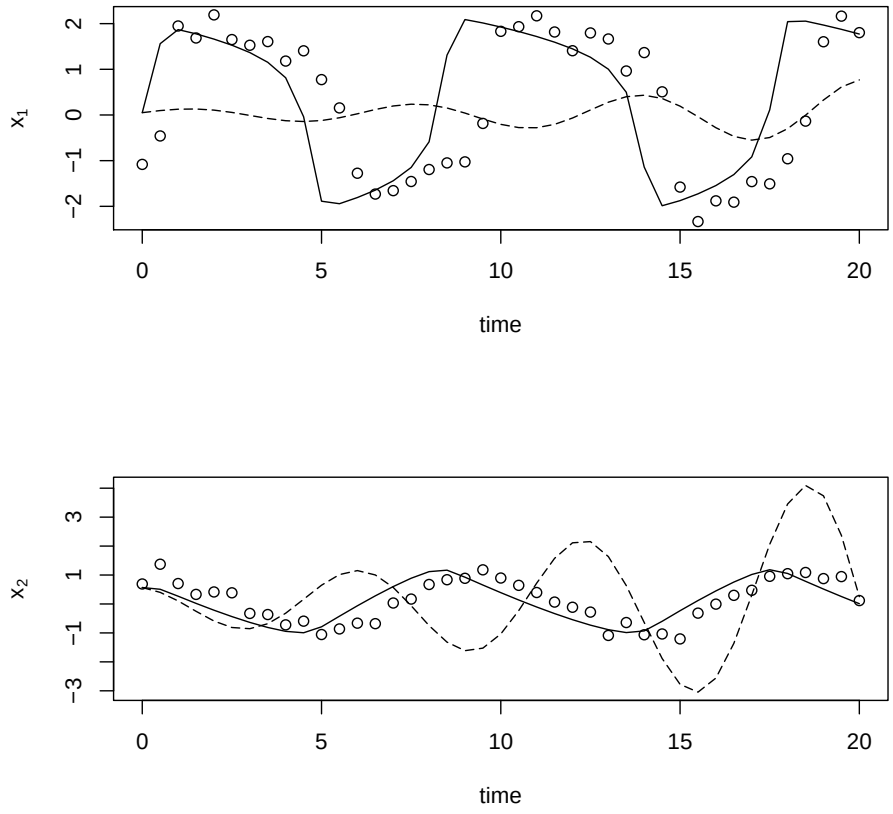



Fig. 4.2 *FhNdata*. The data are represented by the circles. The curves are obtained by solving the FitzHugh-Nagumo system for the parameter $(\widehat{a}_1, \widehat{b}_1, \widehat{c}_1)^\top = (0.017, -0.007, 0.160)^\top$ (dashed lines) and $(\widehat{a}_2, \widehat{b}_2, \widehat{c}_2)^\top = (0.318, -0.140, 3.003)^\top$ (solid lines) and initial condition $(\widehat{\xi}_1, \widehat{\xi}_2)^\top = (0.051, 0.569)^\top$.

| Estimation of parameters in Fitz-Hugh Nagumo system | | | | | | |
|---|---|---|---|---|---|---|
| Method | Initial value | $a = 0.2$ | $b = 0.2$ | $c = 3$ | Relative error | Time(secs) |
| Window | NO | 0.318 | -0.140 | 3.003 | 2.291 | 0.011 |
| Generalized profiling | Window | **0.237** | **0.144** | **3.050** | **0.483** | **25.466** |
| | $0.1u$ | 0.239 | 0.009 | 0.069 | 2.130 | 8.687 |
| | $1u$ | 0.144 | 0.567 | 3.461 | 2.266 | 55.725 |
| | $10u$ | 1.797 | 3.082 | 3.295 | 22.493 | 123.914 |
| | $100u$ | 1.829 | 3.142 | 3.251 | 22.935 | 151.203 |
| | $1000u$ | **0.237** | **0.144** | **3.050** | **0.483** | 380.798 |
| | $10000u$ | 6e+99 | 6e+99 | 1e+56 | 6e+100 | 681.449 |

Table 4.3 Fitz-Hugh Nagumo system. The data are obtained from the R package `CollocInfer` and comprise 41 equally spaced observations on the interval $[0, 20]$. The window estimator does not require an initial guess (first row) and as such it can be used as initial guess for the generalized profiling method (second row). Estimates obtained by the generalized profiling method with initial guesses $10^{i-1}u$, $i = 1, \ldots, 5$, where $u = (1, 1, 1)^\top$, are shown in other rows. The best results are boldfaced. Comparison in terms of running time is only between the two best, boldfaced estimates.

## 4.4 Computational complexity

Computing the window smoother costs $O(n)$ flops. The main bottleneck in computing the proposed estimator is the inversion of the matrices $\widehat{\mathbf{B}}_n$ and $\mathbf{I}_n - \widehat{\mathbf{A}}_n \widehat{\mathbf{B}}_n^{-1} \widehat{\mathbf{A}}_n^\top$, which are of order $p$ and $d$, respectively. The algorithmic complexity of the window estimator is

$$O(p^3 + d^3 + \sqrt{n} d p^2)$$

flops; thus cubic in $p$ and $d$, and square root in $n$. The derivation is given in the appendix.

## 4.5 Real data example

In this section we apply our method to infectious disease data from England and Wales taken from the web page http://ms.mcmaster.ca/∼bolker/measdata.html (consulted on 13/4/2014). The data contain weekly case reports of measles in England and Wales from 1948 to 1963. Modelling dynamics of measles over time has been much studied (e.g Finkenstädt and Grenfell (2000); Earn et al. (2000); Stone et al. (2007); Olinky

et al. (2008); He et al. (2010); Hooker et al. (2011)). Here, our goal is just to show the applicability of our method. We consider the SIR model for the epidemic process without a seasonality component (see, e.g.,Huppert et al. (2012)),

$$\begin{aligned} S'(t) &= -\beta S(t)I(t), \\ I'(t) &= \beta S(t)I(t) - \gamma I(t), \end{aligned} \tag{4.13}$$

where $S$ stands for the number of susceptible individuals and $I$ stands for the infectious compartment. An individual is transferred into infectious compartment $I$ at rate $\beta$ and, when recovered, leaves $I$ at rate $\gamma$. For measles modelling, an individual experiences one recovery in about 5 days, so we set $\gamma = 1.4$. The SIR model considered here is the Lotka-Volterra system studied in the previous section, where now two parameters are constrained to be equal and one of the parameters is set to zero. However, in this case we only have observations from state $I(\cdot)$, while the proposed method can be applied only if all the states are observed. Although handling of unobserved states will be part of our future work, this particular case is not difficult to handle by using the proposed method. Indeed, from (4.13) it follows that $S(t) = -I(t) + I(0) - \gamma \int_0^t I(s)\mathrm{d}s + S(0)$, which means that $S$ can be obtained from $I(\cdot)$ and $S(0)$. Here we assume that $I(0)$ is known, which is common practice in the literature on infectious disease. Hence the statistical problem is reduced to estimating $\beta$ and $S(0)$. In this case, we need to optimize with respect to $S(0)$; we started the optimization with an initial guess for $S(0)$ generated from a uniform distribution over $[0, 1 \times 10^7]$. As measles is a childhood disease, we let time start in September when school begins. The results are shown in Figure 4.3. In the upper panel, the estimated $S(\cdot)$ (dashed line) and the solution based on the parameter estimate for $\beta$ (solid line) are plotted. In the lower panel the observations are displayed with circles while the solution based on the window-based estimate for $\beta$ is plotted with a solid smooth line; the window estimator itself is given by the stepwise solid line. The $\beta$ estimate is $3.30 \times 10^{-7}$ while the $S(0)$ estimate is $4.59 \times 10^6$; these estimates are similar to those obtained in Fine and Clarkson (1982).

## 4.6 Discussion

The window estimator is similar to those proposed by Brunel (2008) and Gugushvili and Klaassen (2012). The main difference is that by considering the system of integral equations (4.2) we avoid the estimation of $\boldsymbol{x}'(\cdot)$. This is important because the

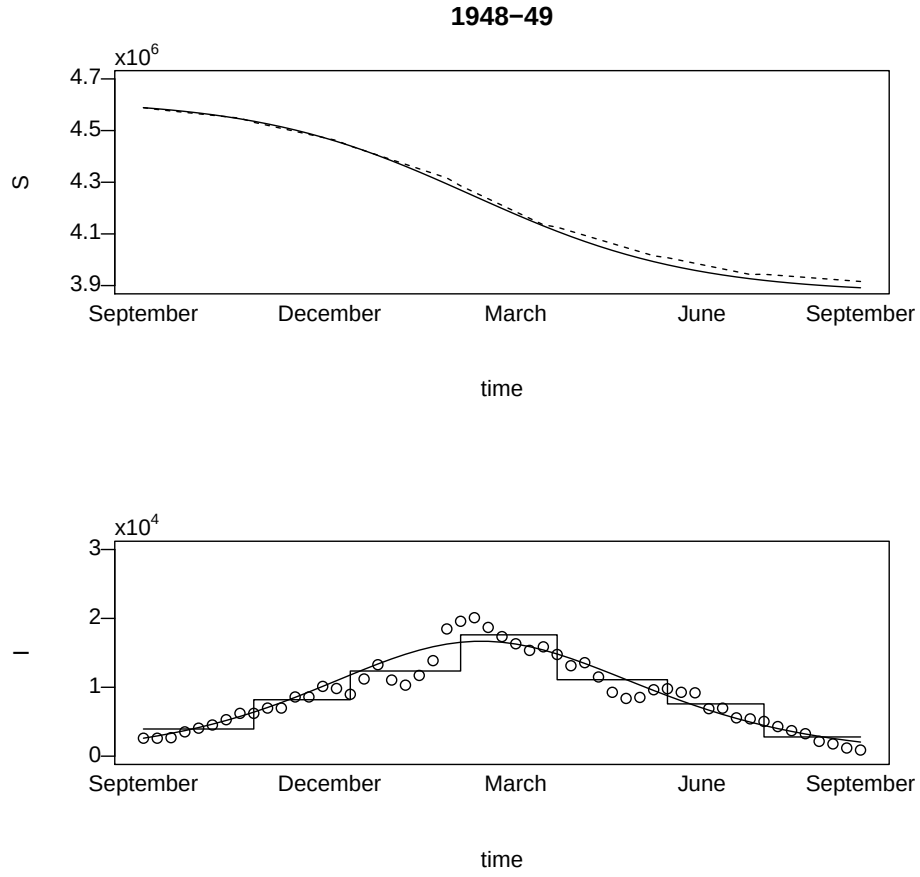



Fig. 4.3 England Wales measles data. Upper panel: The solid line is the solution $S(\cdot)$ based on the parameter estimate $\beta$ and the dashed line is the estimated $S(\cdot)$. Lower panel: The data are given by the circles, the solid line is the solution based on the parameter estimate $\beta$ and the stepwise solid line is the window estimator.

assumption is that we observe only the noisy version of $\boldsymbol{x}(\cdot)$. It is well known that estimating a derivative is less accurate than estimating the curve itself (Loader, 1999, Section 6.1). On the other hand, avoiding the estimation of the derivative induces a new parameter $\boldsymbol{\xi}$. If $\boldsymbol{\xi}$ is identifiable, this poses no problem since we also have an explicit form of its estimator.

The window estimator works by dividing the time interval into several subintervals, over each of which we estimate $\boldsymbol{x}(\cdot)$ by a constant function. This makes our method similar to the *multiple shooting method* (Bock, 1983), which involves solving the system (4.1) over the subintervals with the constraint that the obtained solutions match at the break points. Whereas in multiple shooting the function pieced together by solutions

obtained over each subinterval is continuous, the window smoother is not. Also, in multiple shooting the interval is divided to improve the convergence and numerical stability of the ODEs solver, but here the window smoothing is introduced as a means to obtain fast but $\sqrt{n}$-consistent estimates of the underlying parameters — we are not directly interested in the solution of the ODEs. In order to improve the estimation accuracy, other non-parametric curve estimators could be used. For example, Dattner and Klaassen (2013) use a local polynomial type of estimator. The trade-off is that using more complicated estimators usually involves choosing tuning parameters. Also, the algorithmic complexity would be affected due to the calculation of integrals and repeated integrals in $\widehat{\mathbf{G}}_n(t)$, $\widehat{\mathbf{A}}_n$ and $\widehat{\mathbf{B}}_n$.

## 4.7 Summary

In this chapter, we have presented an extension of a method for parameter estimation in systems of ODEs that are linear in their parameters when the data contain no replicates. Our method is completely automatic; it requires no tuning whatsoever. It is computationally inexpensive and also does not need any initial values of the parameters. This is very appealing, since in Bayesian and likelihood-based methods a good initial estimate is essential. Thus, the method can be used as a fast way to obtain an initial value for the MCMC in the case of Bayesian methods or an initial guess for optimization in the case of likelihood-based methods.

## 4.A Proof of Theorem 1

Denote the number of time points in subinterval $S_i$ by $I_i$. The length of each subinterval $S(t)$ is $T/I$. We have that $\sum_{i=1}^{I} I_i = n$ and $\lfloor\sqrt{n}\rfloor \leq I_i \leq \lfloor\sqrt{n}\rfloor + 2$, which implies

$$\sum_{i=1}^{I} \frac{1}{I_i} = O(1), \tag{4.14}$$

$$\sum_{i=1}^{I} \frac{1}{I_i^3} = O\Big(\frac{1}{n}\Big). \tag{4.15}$$

With the following notation

$$\boldsymbol{\varepsilon}(t_i) = (\varepsilon_1(t_i), \ldots, \varepsilon_d(t_i))^\top,$$

$$\boldsymbol{\varepsilon}(t_i) = \boldsymbol{y}(t_i) - \boldsymbol{x}(t_i),$$

we have

$$\begin{aligned}
\mathrm{E}\varepsilon_k(t_i) &= 0, \qquad k = 1, \ldots, d,\ i = 1, \ldots, n, \\
\mathrm{E}\widehat{\boldsymbol{x}}_n(t) &= \frac{1}{|S(t)|} \sum_{t_j \in S(t)} \boldsymbol{x}(t_j),\ t \in S(t), \\
\widehat{\boldsymbol{x}}_n(t) - \mathrm{E}\widehat{\boldsymbol{x}}_n(t) &= \frac{1}{|S(t)|} \sum_{t_j \in S(t)} \boldsymbol{\varepsilon}(t_j), \qquad (4.16) \\
\mathrm{E}\left\{ \frac{1}{|S(t)|} \sum_{t_j \in S(t)} \varepsilon_k(t_j) \right\} &= 0, \quad k = 1, \ldots, d, \\
\mathrm{E}\{\varepsilon_k(t_i)\varepsilon_k(t_j)\} &= 0, \quad i \neq j, \qquad (4.17)
\end{aligned}$$

and for any $t \in [0, T]$ we have

$$\mathrm{var}\left\{ \frac{1}{|S(t)|} \sum_{t_j \in S(t)} \varepsilon_k(t_j) \right\} = \frac{1}{|S(t)|^2} \mathrm{var}\left\{ \sum_{t_j \in S(t)} \varepsilon_k(t_j) \right\} = \frac{\sigma^2}{|S(t)|}.$$

We use these results throughout the proof. The proof is based on verifying the following conditions as stated in Theorem 2 in (Dattner and Klaassen, 2013).

**1.** $\|\mathrm{E}\widehat{\boldsymbol{x}}_n - \boldsymbol{x}\|_\infty^2 = O(\frac{1}{n})$
For nonnegative numbers $a_1, \ldots, a_m$ it holds that

$$\Big( \sum_{k=1}^{m} a_k \Big)^2 \leq m \sum_{k=1}^{m} a_k^2. \qquad (4.18)$$

The inequality is the consequence of the inequality between arithmetic and quadratic mean. Applying the triangle inequality and (4.18) yields

$$\begin{aligned}
\|\mathrm{E}\widehat{\boldsymbol{x}}_n - \boldsymbol{x}\|_\infty^2 &= \max_{1 \leq i \leq I} \sup_{t \in S_i} \| \frac{1}{I_i} \sum_{t_j \in S_i} \boldsymbol{x}(t_j) - \boldsymbol{x}(t) \|^2 \\
&= \max_{1 \leq i \leq I} \sup_{t \in S_i} \| \frac{1}{I_i} \sum_{t_j \in S_i} \{\boldsymbol{x}(t_j) - \boldsymbol{x}(t)\} \|^2 \\
&\leq \max_{1 \leq i \leq I} \sup_{t \in S_i} \frac{1}{I_i^2} \Big( \sum_{t_j \in S_i} \| \int_t^{t_j} \boldsymbol{x}'(s) \mathrm{d}s \| \Big)^2 \\
&\overset{(4.18)}{\leq} \max_{1 \leq i \leq I} \sup_{t \in S_i} \frac{I}{I_i^2} \sum_{t_j \in S_i} \| \int_t^{t_j} \mathbf{g}(\boldsymbol{x}(s)) \boldsymbol{\theta} \mathrm{d}s \|^2
\end{aligned}$$

$$
\begin{aligned}
&\leq \max_{1\leq i\leq I} \sup_{t\in S_i} \frac{I}{I_i^2} \sum_{t_j\in S_i} \sup_{s\in S_i} \|\mathbf{g}(\boldsymbol{x}(s))\|^2 \|\boldsymbol{\theta}\|^2 (t_j - t)^2 \\
&\leq \max_{1\leq i\leq I} \frac{I}{I_i^2} \sum_{t_j\in S_i} \frac{T^2}{I^2} \sup_{s\in S_i} \|\mathbf{g}(\boldsymbol{x}(s))\|^2 \|\boldsymbol{\theta}\|^2 \\
&\leq \max_{1\leq i\leq I} \frac{I}{I_i^2} \frac{I_i}{I^2} \cdot O(1) = O\Big(\frac{1}{\sqrt{n}\sqrt{n}}\Big) = O\Big(\frac{1}{n}\Big).
\end{aligned}
$$

Here we used boundedness of parameter space $\Theta$. Also, since $\mathbf{g}(\cdot)$ is continuous and $\boldsymbol{x}(\cdot)$ is bounded, then $\mathbf{g}(\boldsymbol{x}(\cdot))$ is bounded.

**2.** $\|\widehat{\boldsymbol{x}}_n\|_\infty = O_p(1)$

$$
\begin{aligned}
P(\|\widehat{\boldsymbol{x}}_n - \mathrm{E}\widehat{\boldsymbol{x}}_n\|_\infty \geq M) &= P(\sup_{t\in[0,T]} \|\widehat{\boldsymbol{x}}_n(t) - \mathrm{E}\widehat{\boldsymbol{x}}_n(t)\| \geq M) \\
&\overset{(4.16)}{=} P(\sup_{t\in[0,T]} \| \frac{1}{|S(t)|} \sum_{t_j\in S(t)} \boldsymbol{\varepsilon}(t_j) \| \geq M) \\
&= P\Big(\max_{1\leq i\leq I} \|\frac{1}{I_i} \sum_{t_j\in S_i} \boldsymbol{\varepsilon}(t_j)\| \geq M\Big) \\
&= 1 - P\Big(\max_{1\leq i\leq I} \|\frac{1}{I_i} \sum_{t_j\in S_i} \boldsymbol{\varepsilon}(t_j)\| \leq M\Big) \\
&= 1 - \prod_{i=1}^{I} P\Big(\|\frac{1}{I_i} \sum_{t_j\in S_i} \boldsymbol{\varepsilon}(t_j)\| \leq M\Big) \\
&= 1 - \prod_{i=1}^{I} \Big\{1 - P\Big(\|\frac{1}{I_i} \sum_{t_j\in S_i} \boldsymbol{\varepsilon}(t_j)\| \geq M\Big)\Big\} \\
&\leq 1 - \prod_{i=1}^{I} \Big(1 - \frac{d\sigma^2}{M^2 I_i}\Big) \\
&\leq 1 - \Big(1 - \sum_{i=1}^{I} \frac{d\sigma^2}{M^2} \frac{1}{I_i}\Big) \leq \frac{d\sigma^2}{M^2} \sum_{i=1}^{I} \frac{1}{I_i}.
\end{aligned}
$$

The first inequality above follows from Chebyshev's inequality and the second from Lemma 4.1, which is given in the Appendix 4.B. We conclude that $\widehat{\boldsymbol{x}}_n - \mathrm{E}\widehat{\boldsymbol{x}}_n$ is bounded in probability. Since we have

$$
\widehat{\boldsymbol{x}}_n = \underbrace{\widehat{\boldsymbol{x}}_n - \mathrm{E}\widehat{\boldsymbol{x}}_n}_{O_p(1)} + \underbrace{\mathrm{E}\widehat{\boldsymbol{x}}_n - \boldsymbol{x}}_{O(\frac{1}{n})} + \underbrace{\boldsymbol{x}}_{O(1)},
$$

it follows that $\widehat{\boldsymbol{x}}_n(\cdot)$ is bounded in probability, i.e. $\|\widehat{\boldsymbol{x}}_n\|_\infty = O_p(1)$.

**3.** $\mathrm{E}(\int_0^T \|\widehat{\boldsymbol{x}}_n(t) - \mathrm{E}\widehat{\boldsymbol{x}}_n(t)\|^2 \mathrm{d}t) = O(\frac{1}{\sqrt{n}})$

$$
\begin{aligned}
\mathrm{E}\Big(\int_0^T \|\widehat{\boldsymbol{x}}_n(t) - \mathrm{E}\widehat{\boldsymbol{x}}_n(t)\|^2 \mathrm{d}t\Big) &\overset{(4.16)}{=} \mathrm{E}\Big(\sum_{i=1}^{I}\int_{S_i} \|\frac{1}{|S(t)|}\sum_{t_j\in S(t)} \boldsymbol{\varepsilon}(t_j)\|^2 \mathrm{d}t\Big) \\
&= \mathrm{E}\Big(\frac{1}{I}\sum_{i=1}^{I} \|\frac{1}{I_i}\sum_{t_j\in S_i} \boldsymbol{\varepsilon}(t_j)\|^2\Big) \\
&= \frac{1}{I}\sum_{i=1}^{I}\frac{1}{I_i^2}\mathrm{E}\Big[\sum_{k=1}^{d}\Big\{\sum_{t_j\in S_i}\varepsilon_k(t_j)\Big\}^2\Big] \\
&= \frac{1}{I}\sum_{i=1}^{I}\frac{1}{I_i^2}\sum_{k=1}^{d}\sum_{t_j\in S_i}\mathrm{E}\varepsilon_k(t_j)^2 \\
&+ \frac{1}{I}\sum_{i=1}^{I}\frac{1}{I_i^2}\sum_{k=1}^{d}\sum_{t_j,t_l\in S_i: j\neq l}\mathrm{E}\varepsilon_k(t_j)\varepsilon_k(t_l) \\
&\overset{(4.17)}{=} \frac{1}{I}\sum_{i=1}^{I}\frac{1}{I_i^2}\sum_{k=1}^{d} I_i\sigma^2 = \frac{1}{I}\sum_{i=1}^{I}\frac{d\sigma^2}{I_i} \\
&\overset{(4.14)}{=} O\Big(\frac{1}{I}\Big) = O\Big(\frac{1}{\sqrt{n}}\Big).
\end{aligned}
$$

**4.** $\int_0^T \mathrm{var}\{\int_0^t \phi(s)\widehat{x}_{n,k}(s)ds\}\mathrm{d}t = O(\frac{1}{n})$

By using Lemma 4.2 (Appendix 4.B) we obtain

$$
\begin{aligned}
\int_0^T \mathrm{var}\left\{\int_0^t \phi(s)\widehat{x}_{n,k}(s)\mathrm{d}s\right\}\mathrm{d}t &= \sum_{i=1}^{I}\int_{S_i}\mathrm{var}\left\{\int_0^t \phi(s)\widehat{x}_{n,k}(s)\mathrm{d}s\right\}\mathrm{d}t \\
&= \sum_{i=1}^{I}\int_{S_i}\Big[\sum_{l=1}^{I}\Big\{\int_{S_l\cap[0,t]}\phi(s)\mathrm{d}s\Big\}^2\frac{\sigma^2}{I_l}\Big]\mathrm{d}t \\
&\leq \|\phi\|_\infty^2\sum_{i=1}^{I}\int_{S_i}\mathrm{d}t\sum_{l=1}^{I}\frac{\sigma^2}{I_l^3} \\
&= \|\phi\|_\infty^2\sum_{i=1}^{I}\frac{1}{I_i}\sum_{l=1}^{I}\frac{\sigma^2}{I_l^3} \\
&= O\Big(\frac{1}{n}\Big),
\end{aligned}
$$

where the last equality is due to (4.14) and (4.15).

**5.** $\mathrm{var}\{\int_0^T \phi(t)\widehat{x}_{n,k}(t)\mathrm{d}t\} = O(\frac{1}{n})$

According to Lemma 4.2 (Appendix 4.B) for any $t \in [0,T]$

$$\operatorname{var}\left\{\int_0^t \phi(s)\widehat{x}_{n,k}(s)\mathrm{d}s\right\} = \sum_{i=1}^{I}\left\{\int_{S_i\cap[0,t]}\phi(s)\mathrm{d}s\right\}^2\frac{\sigma^2}{I_i},$$

whence it follows that

$$\begin{aligned}
\operatorname{var}\left\{\int_0^T \phi(t)\widehat{x}_{n,k}(t)\mathrm{d}t\right\} &= \sum_{i=1}^{I}\left\{\int_{S_i\cap[0,T]}\phi(t)\mathrm{d}t\right\}^2\frac{\sigma^2}{I_i} \\
&= \sum_{i=1}^{I}\left\{\int_{S_i}\phi(t)\mathrm{d}t\right\}^2\frac{\sigma^2}{I_i} \\
&\leq \sum_{i=1}^{I}\left(\|\phi\|_\infty\frac{1}{I}\right)^2\frac{\sigma^2}{I_i} \\
&= \|\phi\|_\infty^2\frac{\sigma^2}{I^2}\sum_{i=1}^{I}\frac{1}{I_i} \\
&\overset{(4.14)}{=} O\Big(\frac{1}{I^2}\Big) = O\Big(\frac{1}{n}\Big).
\end{aligned}$$

## 4.B Auxiliary results

**Lemma 4.1.** *For any $0 \leq a_1, \ldots, a_n \leq 1$ the following inequality holds*

$$\prod_{i=1}^{n}(1-a_i) \geq 1 - \sum_{i=1}^{n} a_i.$$

*Proof.* Using induction. □

**Lemma 4.2.** *For any $t \in [0,T]$ the following equality holds*

$$\operatorname{var}\left\{\int_0^t \phi(s)\widehat{x}_{n,k}(s)\mathrm{d}s\right\} = \sum_{i=1}^{I}\left\{\int_{S_i\cap[0,t]}\phi(s)\mathrm{d}s\right\}^2\frac{\sigma^2}{I_i}.$$

*Proof.* Let $C_i$ denote $\widehat{x}_{n,k}(S_i)$. Since, $[0,T] = \cup_{i=1}^{I} S_i$ then for any $t \in [0,T]$ we have that $[0,t] = \cup_{i=1}^{I}(S_i \cap [0,t])$. Now we have

$$\begin{aligned}
\operatorname{var}\left\{\int_0^t \phi(s)\widehat{x}_{n,k}(s)\mathrm{d}s\right\} &= \operatorname{var}\left\{\sum_{i=1}^{I}\int_{S_i\cap[0,t]}\phi(s)\widehat{x}_{n,k}(s)\mathrm{d}s\right\} \\
&= \operatorname{var}\left\{\sum_{i=1}^{I}\int_{S_i\cap[0,t]}\phi(s)\mathrm{d}s\cdot C_i\right\}
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{i=1}^{I} \left\{ \int_{S_i \cap [0,t]} \phi(s) \mathrm{d}s \right\}^2 \mathrm{var}(C_i) \\
&+ 2 \sum_{1 \leq m < l \leq I} \int_{S_m \cap [0,t]} \phi(s) \mathrm{d}s \int_{S_l \cap [0,t]} \phi(s) \mathrm{d}s \cdot \mathrm{cov}(C_m, C_l),
\end{aligned}
$$

where in the second equality we have used the fact that $\widehat{x}_{n,k}$ is constant on $S_i$ and equal to $C_i$. We need $\mathrm{var}(C_i)$ and $\mathrm{cov}(C_m, C_l)$.

$$
\mathrm{var}(C_i) = \mathrm{var} \left\{ \frac{1}{I_i} \sum_{t_j \in S_i} y_k(t_j) \right\} = \frac{1}{I_i^2} I_i \sigma^2 = \frac{\sigma^2}{I_i},
$$

$$
\begin{aligned}
\mathrm{cov}(C_m, C_l) &= \mathrm{cov} \left( \frac{1}{I_m} \sum_{t_i \in S_m} y_k(t_i), \frac{1}{I_l} \sum_{t_j \in S_l} y_k(t_j) \right) \\
&= \frac{1}{I_m I_l} \sum_{t_i \in S_m, t_j \in S_l} \mathrm{cov}(y_k(t_i), y_k(t_j)) = 0.
\end{aligned}
$$

Plugging the variance and covariance terms into the original expression we obtain the formula. □

## 4.C Calculation of the algorithmic complexity

The following results are taken from Wood and Lindgren (2013) and are used in the sequel.

**Lemma 4.3.** *If* $\mathbf{A}$ *and* $\mathbf{B}$ *are matrices of dimension* $n \times m$ *then the computational cost of addition and subtraction i.e.,* $\mathbf{A} + \mathbf{B}$, $\mathbf{A} - \mathbf{B}$, *is* $\mathcal{O}(mn)$ *flops.*

**Lemma 4.4.** *If* $\mathbf{A}$ *is a matrix of dimension* $n \times m$ *and* $\mathbf{B}$ *is a matrix of* $m \times p$ *then the computational cost of the matrix multiplication* $\mathbf{AB}$ *is* $\mathcal{O}(nmp)$ *flops.*

**Lemma 4.5.** *If a matrix* $\mathbf{A}$ *is of order* $n$ *then the computational cost of the matrix inversion* $\mathbf{A}^{-1}$ *is* $O(n^3)$ *flops.*

The matrices that appear in the form for the estimators have the following dimensions:

- $\mathbf{I}_d$ and $\widehat{\mathbf{A}}_n \widehat{\mathbf{B}}_n^{-1} \widehat{\mathbf{A}}_n^\top$ are of dimension $d \times d$.
- $\widehat{\mathbf{A}}_n$, $\widehat{\mathbf{G}}_n(t_i)$, $\widehat{\mathbf{A}}_n \widehat{\mathbf{B}}_n^{-1}$ and $\mathbf{g}(\widehat{\boldsymbol{x}}_n(t))$ are of dimension $d \times p$.
- $\widehat{\mathbf{A}}_n^\top$ and $\widehat{\mathbf{G}}_n^\top(t_i)$ are of dimension $p \times d$.

- $\widehat{\mathbf{B}}_n$ is of dimension $p \times p$.
- $\widehat{\boldsymbol{x}}_n(t_i)$, $\boldsymbol{y}(t_i)$ and $\widehat{\boldsymbol{\xi}}_n$ are of dimension $d \times 1$.

**1. The cost for $\widehat{\boldsymbol{x}}_n(S_i)$ and $\widehat{\mathbf{G}}_n(a_i)$**

We calculate $\widehat{\boldsymbol{x}}_n(S_i)$, for $i = 1, \ldots, I$. The cost for $\widehat{\boldsymbol{x}}_n(S_i)$ is $O((I-1)d)$ so the total cost is $O(I(I-1)d) = O(I^2 d) = O(nd)$. The calculation of $\widehat{\mathbf{G}}_n(a_i)$ involves the calculation of $\mathbf{g}(\widehat{\boldsymbol{x}}_n(S_i))$ and since the construction of matrix $\mathbf{g}$ depends on the form of the ODEs we assume that every entry of the matrix is obtained by at most a constant number of flops $C$ that does not depend on $n$, $p$ and $d$. Then the cost of obtaining matrix $\mathbf{g}(\widehat{\boldsymbol{x}}_n(S_i))$, for any $i = 1, \ldots, I$, is $O(Cdp) = O(dp)$ flops . The estimator $\widehat{\mathbf{G}}_n(t)$ can be calculated recursively by

$$\begin{aligned} \widehat{\mathbf{G}}_n(a_1) &= \mathbf{g}(\widehat{\boldsymbol{x}}_n(S_1))a_1, \\ \widehat{\mathbf{G}}_n(a_i) &= \widehat{\mathbf{G}}_n(a_{i-1}) + \mathbf{g}(\widehat{\boldsymbol{x}}_n(S_i))\Delta, \qquad i = 2, \ldots, I. \end{aligned} \tag{4.19}$$

Since the cost for computing $\mathbf{g}(\widehat{\boldsymbol{x}}_n(S_i))$, for $i = 1, \ldots, I$ is $O(dp)$ flops, from (4.19) it follows that $\widehat{\mathbf{G}}_n(a_i)$, $i = 1, \ldots, I$ can be computed in $O(dp)$ flops and therefore summing across $i = 1, \ldots, I$ we obtain that the total cost is $O(Idp) = O(\sqrt{n}dp)$ flops.

**2. The cost for $\widehat{\mathbf{A}}_n$, $\widehat{\mathbf{A}}_n(i)$ and $\widehat{B_n}$**

Since the cost for computing $\mathbf{g}(\widehat{\boldsymbol{x}}_n(S_i))$ and $\widehat{\mathbf{G}}_n(a_i)$, for each $i$ is $O(dp)$ flops, from (4.9) it follows that the cost for $\widehat{\mathbf{A}}_n(i)$ is $O(dp)$ flops. The equality $\widehat{\mathbf{A}}_n = \sum_{i=1}^{n} \widehat{\mathbf{A}}_n(i)$ implies that the total cost for computing $\widehat{\mathbf{A}}_n$ and $\widehat{\mathbf{A}}_n(i)$, $i = 1, \ldots, n$ is $O(Idp)$ flops. For the cost of $\widehat{\mathbf{B}}_n$ we analyze each term in (4.10).

- $(\widehat{\mathbf{G}}_n(T)^\top, \widehat{\mathbf{A}}_n) \mapsto \widehat{\mathbf{G}}_n(T)^\top \widehat{\mathbf{A}}_n$ costs $O(p^2 d)$ flops.
- $\left(\mathbf{g}(\widehat{\boldsymbol{x}}_n(S_i))^\top, t^2 \widehat{\mathbf{G}}_n(t)\Big|_{a_{i-1}}^{a_i}\right) \mapsto \mathbf{g}(\widehat{\boldsymbol{x}}_n(S_i))^\top t^2 \widehat{\mathbf{G}}_n(t)\Big|_{a_{i-1}}^{a_i}$ costs $O(p^2 d)$ flops.
- $(\mathbf{R}_i^\top, \mathbf{g}(\widehat{\boldsymbol{x}}_n(S_i))(2i-1)) \mapsto \mathbf{R}_i^\top \mathbf{g}(\widehat{\boldsymbol{x}}_n(S_i))(2i-1)$ costs $O(p^2 d)$ flops.
- $(\mathbf{g}(\widehat{\boldsymbol{x}}_n(S_i))^\top, \mathbf{g}(\widehat{\boldsymbol{x}}_n(S_i))(3i^2-1)) \mapsto \mathbf{g}(\widehat{\boldsymbol{x}}_n(S_i))^\top \mathbf{g}(\widehat{\boldsymbol{x}}_n(S_i))(3i^2-1)$ costs $O(p^2 d)$ flops.

Each term above has complexity $O(p^2 d)$ and summing over $i$ we get that the cost for $\widehat{\mathbf{B}}_n$ is $O(Ip^2 d)$. Finally, the total cost for $\widehat{\mathbf{A}}_n$, $\widehat{\mathbf{A}}_n(i)$, $i = 1, \ldots, I$, and $\widehat{\mathbf{B}}_n$ is $O(Idp) + O(Ip^2 d) = O(Ip^2 d)$, which is equal to $O(\sqrt{n}p^2 d)$ flops.

**3. The cost for $\widehat{\boldsymbol{\xi}}_n$**

**The term $\left(T\mathbf{I}_d - \widehat{\mathbf{A}}_n \widehat{\mathbf{B}}_n^{-1} \widehat{\mathbf{A}}_n^\top\right)^{-1}$**

- $\widehat{\mathbf{B}}_n \mapsto \widehat{\mathbf{B}}_n^{-1}$ costs $O(p^3)$ flops.
- $(\widehat{\mathbf{A}}_n, \widehat{\mathbf{B}}_n^{-1}) \mapsto \widehat{\mathbf{A}}_n \widehat{\mathbf{B}}_n^{-1}$ costs $O(p^2 d)$ flops.
- $(\widehat{\mathbf{A}}_n \widehat{\mathbf{B}}_n^{-1}, \widehat{\mathbf{A}}_n^\top) \mapsto \widehat{\mathbf{A}}_n \widehat{\mathbf{B}}_n^{-1} \widehat{\mathbf{A}}_n^\top$ costs $O(d^2 p)$ flops.

- $(\mathbf{I}_d, \widehat{\mathbf{A}}_n\widehat{\mathbf{B}}_n^{-1}\widehat{\mathbf{A}}_n^\top) \mapsto T\mathbf{I}_d - \widehat{\mathbf{A}}_n\widehat{\mathbf{B}}_n^{-1}\widehat{\mathbf{A}}_n^\top$ costs $O(d^2)$ flops.
- $\mathbf{I}_d - \widehat{\mathbf{A}}_n\widehat{\mathbf{B}}_n^{-1}\widehat{\mathbf{A}}_n^\top \mapsto (\mathbf{I}_d - \widehat{\mathbf{A}}_n\widehat{\mathbf{B}}_n^{-1}\widehat{\mathbf{A}}_n^\top)^{-1}$ costs $O(d^3)$ flops.

Thus, the computational cost of $\left(\mathbf{I}_d - \widehat{\mathbf{A}}_n\widehat{\mathbf{B}}_n^{-1}\widehat{\mathbf{A}}_n^\top\right)^{-1}$ is $O(p^3 + p^2d + d^2p + d^2 + d^3)$ which is equal to $O(p^3 + d^3)$ flops.

**The term** $T\sum_{i=1}^I \widehat{\boldsymbol{x}}_n(S_i)/I - \widehat{\mathbf{A}}_n\widehat{\mathbf{B}}_n^{-1}\sum_{i=1}^I \widehat{\mathbf{A}}_n(i)^\top\widehat{\boldsymbol{x}}_n(S_i)$

- $\widehat{\boldsymbol{x}}_n(S_i) \mapsto T\sum_{i=1}^I \widehat{\boldsymbol{x}}_n(S_i)/I$ costs $O(Id)$ flops.
- $(\widehat{\mathbf{A}}_n(i)^\top, \widehat{\boldsymbol{x}}_n(S_i)) \mapsto \sum_{i=1}^I \widehat{\mathbf{A}}_n(i)^\top\widehat{\boldsymbol{x}}_n(S_i)$ costs $O(Ipd)$ flops.
- $(\widehat{\mathbf{A}}_n\widehat{\mathbf{B}}_n^{-1}, \sum_{i=1}^I \widehat{\mathbf{A}}_n(i)^\top\widehat{\boldsymbol{x}}_n(S_i)) \mapsto \widehat{\mathbf{A}}_n\widehat{\mathbf{B}}_n^{-1}\sum_{i=1}^I \widehat{\mathbf{A}}_n(i)^\top\widehat{\boldsymbol{x}}_n(S_i)$ costs $O(dp)$ flops.

In total, the cost for $T\sum_{i=1}^I \widehat{\boldsymbol{x}}_n(S_i)/I - \widehat{\mathbf{A}}_n\widehat{\mathbf{B}}_n^{-1}\sum_{i=1}^I \widehat{\mathbf{A}}_n(i)^\top\widehat{\boldsymbol{x}}_n(S_i)$ is $O(Idp) = O(\sqrt{n}dp)$ flops. Finally we have the following:

- $\left(\mathbf{I}_d - \widehat{\mathbf{A}}_n\widehat{\mathbf{B}}_n^{-1}\widehat{\mathbf{A}}_n^\top\right)^{-1}$ costs $O(p^3 + d^3)$ flops.
- $T\sum_{i=1}^I \widehat{\boldsymbol{x}}_n(S_i)/I - \widehat{\mathbf{A}}_n\widehat{\mathbf{B}}_n^{-1}\sum_{i=1}^I \widehat{\mathbf{A}}_n(i)^\top\widehat{\boldsymbol{x}}_n(S_i)$ costs $O(\sqrt{n}dp)$ flops.
- Multiplication of two aforementioned terms costs $O(d^2)$ flops.

In total, calculation of $\widehat{\boldsymbol{\xi}}_n$ costs $O(p^3 + d^3 + \sqrt{n}dp + d^2)$, which is equal to $O(p^3 + d^3 + \sqrt{n}dp)$ flops.

**4. The cost for $\widehat{\boldsymbol{\theta}}_n$**

**The term** $\sum_{i=1}^I \widehat{\mathbf{A}}_n(i)^\top\widehat{\boldsymbol{x}}_n(S_i) - \widehat{\mathbf{A}}_n^\top\widehat{\boldsymbol{\xi}}_n$

- $(\widehat{\mathbf{A}}_n(i)^\top, \widehat{\boldsymbol{x}}_n(S_i)) \mapsto \sum_{i=1}^I \widehat{\mathbf{A}}_n(i)^\top\widehat{\boldsymbol{x}}_n(S_i)$ costs $O(Ipd)$ flops.
- $(\widehat{\mathbf{A}}_n^\top, \widehat{\boldsymbol{\xi}}_n) \mapsto \widehat{\mathbf{A}}_n^\top\widehat{\boldsymbol{\xi}}_n$ costs $O(pd)$ flops.
- Addition of two aforementioned terms costs $O(p)$ flops.

In total, the cost for calculating $\sum_{i=1}^I \widehat{\mathbf{A}}_n(i)^\top\widehat{\boldsymbol{x}}_n(S_i) - \widehat{\mathbf{A}}_n^\top\widehat{\boldsymbol{\xi}}_n$ is $O(Ipd) = O(\sqrt{n}pd)$ flops. The multiplication of $\widehat{\mathbf{B}}_n^{-1}$ with the last mentioned term costs $O(p^2)$ flops so the total cost for $\widehat{\boldsymbol{\theta}}_n$ is $O(p^2 + \sqrt{n}pd)$ flops.

Hence, the computational complexity of the procedure is $O(\sqrt{n}dp + \sqrt{n}dp^2 + p^3 + d^3 + \sqrt{n}dp + p^2 + \sqrt{n}pd)$ flops which is equal to $O(p^3 + d^3 + \sqrt{n}dp^2)$ flops.

## 4.D Calculation of the integrals

Let us first recall that $S_i = [a_{i-1}, a_i]$, for $i = 1, \ldots, I-1$ and $S_I = [a_{I-1}, T]$. The length of each subinterval $S_i$ is $\Delta = T/I$ and the boundary points of the subintervals are $a_i = i\Delta$, for $i = 0, \ldots, I$. To shorten notation, we let $G_1(t)$ stand for an arbitrary entry of the matrix $\widehat{\mathbf{G}}_n(t)$. Similarly, let $g_1(\widehat{\boldsymbol{x}}_n(t))$ denote the entry of the matrix $\mathbf{g}(\widehat{\boldsymbol{x}}_n(t))$ that corresponds to $G_1(t)$, i.e $G_1(t) = \int_0^t g_1(\widehat{\boldsymbol{x}}_n(s))ds$. Let $C_i = g_1(\widehat{\boldsymbol{x}}_n(S_i))$,

for $i = 1, \ldots, I$. Using integration by parts yields

$$\begin{aligned}\int_{S_i} G_1(t)\mathrm{d}t = \int_{a_{i-1}}^{a_i} G_1(t)\mathrm{d}t &= tG_1(t)|_{a_{i-1}}^{a_i} - \int_{a_{i-1}}^{a_i} tg_1(\widehat{\boldsymbol{x}}_n(t))\mathrm{d}t \\ &= tG_1(t)|_{a_{i-1}}^{a_i} - g_1(C_i)(a_i^2 - a_{i-1}^2)/2 \\ &= tG_1(t)|_{a_{i-1}}^{a_i} - \frac{1}{2}(2i-1)\Delta^2 g_1(C_i), \end{aligned} \tag{4.20}$$

where we have used the identity $a_i^2 - a_{i-1}^2 = (2i-1)\Delta^2$. In the same manner, we obtain that for $t \in S_i$ it holds that

$$\begin{aligned}\int_0^t G_1(s)\mathrm{d}s &= sG_1(s)|_0^t - \int_0^t sg_1(\widehat{\boldsymbol{x}}_n(s))\mathrm{d}s \\ &= tG_1(t) - \sum_{m=1}^{i-1}\int_{a_{m-1}}^{a_m} sg_1(\widehat{\boldsymbol{x}}_n(s))\mathrm{d}s - \int_{a_{i-1}}^{t} sg_1(\widehat{\boldsymbol{x}}_n(s))\mathrm{d}s \\ &= tG_1(t) - \sum_{m=1}^{i-1}\frac{g_1(C_m)}{2}(a_m^2 - a_{m-1}^2) - \frac{1}{2}g_1(C_i)(t^2 - a_{i-1}^2) \\ &= tG_1(t) - \frac{\Delta^2}{2}\sum_{m=1}^{i-1}(2m-1)g_1(C_m) - \frac{1}{2}g_1(C_i)(t^2 - a_{i-1}^2), \end{aligned} \tag{4.21}$$

Recall the convention that $\sum_{m=1}^{i-1}(2m-1)g_1(C_m)$ is equal to zero for $i = 1$. Setting $t = T$ in (4.21) we obtain

$$\int_0^T G_1(t)\mathrm{d}t = TG_1(T) - \frac{\Delta^2}{2}\sum_{m=1}^{I}(2m-1)g_1(C_m). \tag{4.22}$$

Applying (4.20) and (4.22) to every entry of the matrices $\int_{S_i}\widehat{\mathbf{G}}_n(t)\mathrm{d}t$ and $\int_0^T \widehat{\mathbf{G}}_n(t)\mathrm{d}t$, respectively, we conclude that

$$\widehat{\mathbf{A}}_n(i) = \int_{S_i}\widehat{\mathbf{G}}_n(t)\mathrm{d}t = t\widehat{\mathbf{G}}_n(t)|_{a_{i-1}}^{a_i} - \frac{1}{2}(2i-1)\Delta^2\mathbf{g}(\widehat{\boldsymbol{x}}_n(S_i)),$$

$$\widehat{\mathbf{A}}_n = \int_0^T \widehat{\mathbf{G}}_n(t)\mathrm{d}t = T\widehat{\mathbf{G}}_n(T) - \frac{\Delta^2}{2}\sum_{i=1}^{I}(2i-1)\mathbf{g}(\widehat{\boldsymbol{x}}_n(S_i)).$$

To obtain $\widehat{\mathbf{B}}_n$, we need integrals of the form $\int_0^T G_1(t)G_2(t)\mathrm{d}t$ where, to simplify notation, entries of the matrix $\widehat{\mathbf{G}}_n(t)$ are denoted by $G_1(t)$ and $G_2(t)$. Similarly $g_1(\widehat{\boldsymbol{x}}_n(t))$ and $g_2(\widehat{\boldsymbol{x}}_n(t))$ denote the corresponding entries of the matrix $\mathbf{g}(\widehat{\boldsymbol{x}}_n(t))$, i.e

$G_i(t) = \int_0^t g_i(\widehat{\boldsymbol{x}}_n(s))ds$, $i = 1, 2$. Integration by parts yields

$$\begin{aligned}\int_0^T G_1(t)G_2(t)\mathrm{d}t &= G_1(s)\int_0^t G_2(s)\mathrm{d}s\Big|_0^T - \int_0^T \Big\{g_1(\widehat{\boldsymbol{x}}_n(s))\int_0^t G_2(s)\mathrm{d}s\Big\}\mathrm{d}t \\ &= G_1(T)\int_0^T G_2(t)\mathrm{d}t - \sum_{i=1}^{I} g_1(C_i)\int_{a_{i-1}}^{a_i}\Big\{\int_0^t G_2(s)\mathrm{d}s\Big\}\mathrm{d}t.\end{aligned}$$

According to (4.21), for any $t \in S_i$

$$\int_0^t G_2(s)\mathrm{d}s = tG_2(t) - \frac{\Delta^2}{2}\sum_{m=1}^{i-1}(2m-1)g_2(C_m) - \frac{g_2(C_i)}{2}(t^2 - a_{i-1}^2)$$

and hence

$$\begin{aligned}\int_{a_{i-1}}^{a_i}\Big\{\int_0^t G_2(s)\mathrm{d}s\Big\}\mathrm{d}t &= \int_{a_{i-1}}^{a_i} tG_2(t)\mathrm{d}t - \frac{\Delta^3}{2}\sum_{m=1}^{i-1}(2m-1)g_2(C_m) \\ &- \frac{g_2(C_i)}{2}\Big\{\frac{1}{3}(a_i^3 - a_{i-1}^3) - a_{i-1}^2(a_i - a_{i-1})\Big\}.\end{aligned}$$

Again using integration by parts, we obtain

$$\begin{aligned}\int_{a_{i-1}}^{a_i} tG_2(t)\mathrm{d}t &= \frac{1}{2}t^2G_2(t)\Big|_{a_{i-1}}^{a_i} - \frac{1}{2}\int_{a_{i-1}}^{a_i} t^2 g_2(\widehat{\boldsymbol{x}}_n(t))\mathrm{d}t \\ &= \frac{1}{2}t^2G_2(t)\Big|_{a_{i-1}}^{a_i} - \frac{1}{2}g_2(C_i)\frac{1}{3}(a_i^3 - a_{i-1}^3).\end{aligned}$$

Since $a_i = i\Delta$, for $i = 0, \dots I$ it follows that

$$a_i^3 - a_{i-1}^3 = (3i^2 - 3i + 1)\Delta^3,$$

$$\frac{1}{3}(a_i^3 - a_{i-1}^3) - a_{i-1}^2(a_i - a_{i-1}) = \Delta^3(3i-2)/3.$$

Also, the following identity holds

$$\sum_{i=1}^{I} g_1(C_i)\sum_{m=1}^{i-1}(2m-1)g_2(C_m) = \sum_{i=1}^{I-1}(2i-1)R_i g_2(C_i),$$

where $R_i = \sum_{m=i+1}^{I} g_1(C_m)$. By using previous identities, we finally obtain

$$\int_0^T G_1(t)G_2(t)\mathrm{d}t = G_1(T)\int_0^T G_2(s)\mathrm{d}s - \frac{1}{2}\sum_{i=1}^{I} g_1(C_i)t^2G_2(t)\Big|_{a_{i-1}}^{a_i}$$

$$
\begin{aligned}
&+ \quad \frac{\Delta^3}{6}\sum_{i=1}^{I}(3i^2-1)g_1(C_i)g_2(C_i) \\
&+ \quad \frac{\Delta^3}{2}\sum_{i=1}^{I-1}(2i-1)R_i g_2(C_i).
\end{aligned}
$$

Since $(r,k)$th entry, $\widehat{\mathbf{G}}_n(t)[r,k]$, of the matrix $\int_0^T \widehat{\mathbf{G}}_n(t)^\top \widehat{\mathbf{G}}_n(t)\mathrm{d}t$ is the sum of expressions $\int_0^T \widehat{\mathbf{G}}_n(t)[l,r]\widehat{\mathbf{G}}_n(t)[l,k]\mathrm{d}t$ that are of the form above we obtain that

$$
\begin{aligned}
\widehat{\mathbf{B}}_n = \int_0^T \widehat{\mathbf{G}}_n(t)^\top \widehat{\mathbf{G}}_n(t)\mathrm{d}t \quad &= \quad \widehat{\mathbf{G}}_n(T)^\top \int_0^T \widehat{\mathbf{G}}_n(t)\mathrm{d}t \\
&- \quad \frac{1}{2}\sum_{i=1}^{I}\mathbf{g}(\widehat{\boldsymbol{x}}_n(S_i))^\top t^2 \widehat{\mathbf{G}}_n(t)\Big|_{a_{i-1}}^{a_i} \\
&+ \quad \frac{\Delta^3}{6}\sum_{i=1}^{I}(3i^2-1)\mathbf{g}^\top(\widehat{\boldsymbol{x}}_n(S_i))\mathbf{g}(\widehat{\boldsymbol{x}}_n(S_i)) \\
&+ \quad \frac{\Delta^3}{2}\sum_{i=1}^{I-1}(2i-1)\mathbf{R}_i^\top \mathbf{g}(\widehat{\boldsymbol{x}}_n(S_i)),
\end{aligned}
$$

where $\mathbf{R}_i = \sum_{m=i+1}^{I}\mathbf{g}(\widehat{\boldsymbol{x}}_n(S_m))$.

# Chapter 5

# RKHS approach to estimating parameters in ordinary differential equations

In this chapter, we discuss the problem of estimating parameters in ODEs which have a general form. In our approach we combine the frequentist set-up, such as in Ramsay et al. (2007), with the kernel approach of Steinke and Schölkopf (2008). In this way we define the estimation problem explicitly as a maximum likelihood problem, where the differential equation is interpreted as a constraint. By introducing a reproducing kernel Hilbert space (RKHS), we transform the constrained maximization problem into an unconstrained one. We detail this idea in Section 5.3 after a review of RKHS and Green's functions given in Section 5.1, and the classical MLE approach reviewed in Section 5.2. In Section 5.4 we focus on the implementation of our methodology. Sections 5.5 and 5.6 illustrate the behaviour of the technique in simulated and real data scenarios, respectively. The last section contains the summary and the appendix contains proofs.

## 5.1 Preliminaries

### 5.1.1 Reproducing kernel Hilbert spaces

In this section we give an introduction to the theory of reproducing kernel Hilbert spaces. Our exposition is based on Steinwart and Christmann (2008). We restrict ourselves to real Hilbert spaces although the theory holds for complex Hilbert spaces

as well.

**Definition 5.1.** *Let $X$ be a nonempty set. A function $k : X \times X \to \mathbb{R}$ is called a kernel on $X$ if it is symmetric and positive definite, i.e. for all $n \in N$, $\alpha_1, \ldots, \alpha_n \in \mathbb{R}$ and all $x_1, \ldots, x_n$ we have*

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j k(x_i, x_j) \geq 0. \tag{5.1}$$

Condition (5.1) is equivalent to the positive definiteness of the *kernel matrix* (or *Gram matrix*)

$$\mathbf{K} = (k(x_i, x_j))_{i,j}. \tag{5.2}$$

Kernels are intimately connected to Hilbert spaces since it can be shown that $k$ is a kernel on $X$ if and only if there exists a Hilbert space $\mathcal{H}$ called a *feature space* and a map $\Phi : X \to \mathcal{H}$ called a *feature map* such that for all $x, x' \in X$ we have

$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle_{\mathcal{H}},$$

where $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ is an inner product on $\mathcal{H}$. For a kernel neither the feature map nor the feature space are uniquely determined. Although not unique, it can be shown that the feature space $\mathcal{H}$ has to be a Hilbert function space over $X$, i.e., a Hilbert space that consists of functions mapping $X$ into $\mathbb{R}$. Related to this, we can consider an opposite situation: for a given Hilbert function space $\mathcal{H}$ over $X$ does there exist a kernel $k$ that can reproduce the functions of the space $\mathcal{H}$? In this regard, we have the following definition.

**Definition 5.2.** *Let $X$ be a nonempty set and $\mathcal{H}$ be a Hilbert function space over $X$. A function $k : X \times X \to \mathbb{R}$ is called a reproducing kernel of $\mathcal{H}$ if we have $k(\cdot, x) \in \mathcal{H}$ for all $x \in X$ and the reproducing property*

$$f(x) = \langle f, k(\cdot, x) \rangle$$

*holds for all $f \in \mathcal{H}$ and all $x \in X$.*

In definition 5.2 it is not assumed that $k$ is a kernel, since it can be shown that every reproducing kernel is a kernel in the sense of definition 5.1 with the feature map

$$\Phi(x) = k(\cdot, x), \qquad x \in X.$$

Furthermore, it can be shown that every Hilbert function space with a reproducing kernel is a reproducing kernel Hilbert space in the sense of the following definition.

**Definition 5.3.** *Let $X$ be a nonempty set. A Hilbert function space $\mathcal{H}$ over $X$ is a reproducing kernel Hilbert space (RKHS) over $X$ if for all $x \in X$ the Dirac functional $\delta_x : \mathcal{H} \to \mathbb{R}$ defined by*

$$\delta_x(f) = f(x), \qquad f \in \mathcal{H},$$

*is continuous.*

There is a one-to-one relation between kernels and RKHSs; that is the content of the following theorem.

**Theorem 5.1.** *1. (Every RKHS has a unique reproducing kernel) Let $\mathcal{H}$ be an RKHS over $X$. Then $k : X \times X \to \mathbb{R}$ defined by*

$$k(x, x') = \langle \delta_x, \delta_{x'} \rangle, \qquad x \in X,$$

*is the only reproducing kernel of $\mathcal{H}$.*

*2. (Every kernel has a unique RKHS) Let $X$ be a nonempty set and $k$ be a kernel on $X$. Then the metric completion $\mathcal{H}$ of the set*

$$\mathcal{H}_{\text{pre}} = \left\{ \sum_{i=1}^{n} \alpha_i k(\cdot, x_i) : n \in N, \alpha_1, \ldots, \alpha_n \in \mathbb{R}, x_1, \ldots, x_n \in X \right\}$$

*is the only RKHS for which $k$ is a reproducing kernel. Also, for $f = \sum_{i=1}^{n} \alpha_i k(\cdot, x_i) \in \mathcal{H}_{pre}$ we have*

$$\|f\|_{\mathcal{H}}^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j k(x_i, x_j) = \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha},$$

*where $\mathbf{K}$ is the kernel matrix defined in (5.2) and $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)^\top$.*

Consider the problem of estimating a function $f : X \to \mathbb{R}$ from the data $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n) \,|x_i \in X, y_i \in \mathbb{R}\}$. We focus here on the quadratic loss but the problem could be formulated for an arbitrary *loss function* (Smola et al., 2002). Minimizing the functional

$$L(f) = a \sum_{i=1}^{n} \{y_i - f(x_i)\}^2,$$

where $a$ is some positive constant, is an *ill-posed problem* (Chen and Haykin, 2002) unless we restrict the class of functions $f$. One approach is to restrict the class of

functions $f$ to a compact set of an RKHS over $X$. This can be done by regularizing the empirical loss $L$ with a strictly monotone function of $\|f\|^2_{\mathcal{H}}$. In our work, we consider the regularized (penalized) loss

$$L_\lambda(f) = a\sum_{i=1}^{n}\{y_i - f(x_i)\}^2 + \frac{\lambda}{2}\|f\|^2_{\mathcal{H}}. \tag{5.3}$$

The existence of the minimizer of the penalized loss and its representation in terms of kernels is guaranteed by the *representer theorem* in the case of an arbitrary loss function and a penalty which is a strictly monotonic function of the RKHS norm (Smola et al., 2002). In the case of the penalized loss (5.3) the solution is also unique.

**Theorem 5.2** (Representer theorem)**.** *Let $a$ and $\lambda$ be positive constants and let $\mathcal{H}$ be an RKHS over $X$ associated to a kernel $k$. Then for all $x_1, \ldots, x_n \in X$ and all $y_1, \ldots, y_n \in \mathbb{R}$ the minimizer $\widehat{f} \in \mathcal{H}$ of the functional*

$$L_\lambda(f) = a\sum_{i=1}^{n}\{y_i - f(x_i)\}^2 + \frac{\lambda}{2}\|f\|^2_{\mathcal{H}}$$

*is unique and admits a representation of the form*

$$\widehat{f}(x) = \sum_{i=1}^{n}\alpha_i k(x_i, x),$$

*where vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)^\top$ is given by*

$$\boldsymbol{\alpha} = \left(\frac{\lambda}{2a}\mathbf{I}_n + \mathbf{K}\right)^{-1}\boldsymbol{y}$$

*and $\boldsymbol{y} = (y_1, \ldots, y_n)^\top$.*

As a consequence of this theorem we have that $\widehat{\boldsymbol{f}} = (\widehat{f}(x_1), \ldots, \widehat{f}(x_n))^\top = \mathbf{K}\boldsymbol{\alpha}$.

### 5.1.2 Green's function and reproducing kernel Hilbert spaces

In this section we cover the concept of Green's function and its relation to RKHS. In our exposition we define Green's function as a function of two variables, though it can be defined as a distribution, i.e. generalized function. The terminology varies (Griffel, 1985; Nikol'skij, 1992; Kelley and Peterson, 2010; Folland, 1992; Duffy, 2001; Stakgold and Holst, 1979; Barton, 1989; Roach, 1982; Fasshauer, 2012), but roughly speaking,

the main idea is that Green's function is the kernel of the integral operator, which is the inverse of a linear differential operator.

**Definition 5.4.** *Let* $\mathrm{P} = \mathrm{d}^m/\mathrm{d}t^m + \sum_{k=1}^{m-1} \theta_k \mathrm{d}^k/\mathrm{d}t^k$ *be a linear differential operator, where* $\theta_1, \ldots, \theta_{m-1} \in \mathbb{R}$. *Any function* $G(t, s)$ *satisfying*

$$\mathrm{P}_t\{G(t,s)\} = \delta(t-s), \tag{5.4}$$

*where* $\delta$ *is the Dirac delta function, is called a fundamental solution or Green's function of the operator* P.

The subscript $t$ in P is to stress that the differentiation is with respect to the variable $t$. The Dirac $\delta$ function is defined as a "function" such that $\int_{-\infty}^{+\infty} \delta(t-s)f(s)ds = f(t)$ and

$$\delta(t) = \begin{cases} +\infty & \text{if } t = 0 \\ 0 & \text{if } t \neq 0 \end{cases}.$$

As we use the delta function for notational simplicity, equality (5.4) should not be understood in a *distributional sense* (Folland, 1992) but as follows:

- $G(t, s)$ belongs to the class $C^{m-2}[a, b]$ and its $(m-1)$st and $m$th derivative exist and are continuous for $t \neq s$.
- $\mathrm{P}_t\{G(t, s)\} = 0$, for $t \neq s$.
- $(m-1)$st derivative has a unit jump for $t = s$.

Green's function is important because it is used to solve linear differential equations. If we have the differential equation

$$\mathrm{P}g(t) = f(t), \qquad t \in [0, T],$$

and certain conditions the solution $g$ needs to satisfy, like *boundary conditions*, *initial value conditions* or more generally *distributed conditions*, then the solution can be written as

$$g(t) = (\mathrm{P}^{-1}f)(t) = \int_0^T G(t,s)f(s)\mathrm{d}s.$$

Besides (5.4), $G$ also needs to satisfy the conditions that arise from those conditions imposed on the solution $g$. In other words, the inverse operator $\mathrm{P}^{-1}$ of the differential operator P is an integral operator with the *kernel* $G$. For more details see Nikol'skij (1992).

In this thesis we use only differential operators of first order, which we discretize. In other words, we use a difference operator and consequently we are interested in its

Green's function. Thus, we consider a Hilbert space of functions over a finite set $X = \{x_1, \ldots, x_n\}$, i.e. $\mathcal{H} = \{f : f : X \to \mathbb{R}\}$. Here the situation is much simpler, because every function $f \in \mathcal{H}$ is fully described by the vector $\boldsymbol{f} = (f(x_1), \ldots, f(x_n))^\top \in \mathbb{R}^n$, linear operators $\mathrm{P} : \mathcal{H} \to \mathcal{H}$ are isomorphic to matrices and any function $G : X \times X \to \mathbb{R}$ uniquely determines a linear operator $\mathrm{G} : \mathcal{H} \to \mathcal{H}$ through its matrix $\mathbf{G}$, where $(\mathbf{G})_{ij} = G(x_i, x_j)$. Also, for every invertible operator on a finite dimensional space we have the form of the matrix of its inverse. The role of the delta function is taken by the Kronecker delta. For more details see Steinke and Schölkopf (2008). Hence, the definition (5.4) for the difference operator with a matrix $\mathbf{P}$ translates into

$$(\mathbf{PG})(t, s) = \delta(t, s),$$

where $\delta(t, s)$ is the Kronecker delta. In other words, $\mathbf{G}$ is the inverse matrix of the matrix $\mathbf{P}$ of a difference operator, i.e. $\mathbf{G} = \mathbf{P}^{-1}$.

In the following theorem we connect Green's function with RKHS.

**Theorem 5.3.** *Let* $\mathbf{P}$ *be the standard matrix of an invertible linear operator on* $\mathbb{R}^n$*. Define an inner product on* $\mathcal{H}$ *with*

$$\langle \boldsymbol{f}, \boldsymbol{f}' \rangle_{\mathcal{H}} = \langle \mathbf{P}\boldsymbol{f}, \mathbf{P}\boldsymbol{f}' \rangle_{\mathbb{R}^n},$$

*where* $\langle \cdot, \cdot \rangle_{\mathbb{R}^n}$ *is the standard Euclidean inner product in* $\mathbb{R}^n$*. Then* $\mathcal{H}$ *is an RKHS over* $X$ *that admits as a reproducing kernel Green's function of* $\mathbf{P}^\top\mathbf{P}$*, i.e. the function* $G : X \times X \to \mathbb{R}$ *determined by the Green's matrix*

$$\mathbf{G} = (\mathbf{P}^\top\mathbf{P})^{-1}.$$

*Proof.* Green's function $G : X \times X \to \mathbb{R}$ determined by the matrix $\mathbf{G} = (\mathbf{P}^\top\mathbf{P})^{-1}$ is the reproducing kernel of $\mathcal{H}$ since for any function $f \in \mathcal{H}$ we have

$$\begin{aligned} \langle f, G(\cdot, x) \rangle_{\mathcal{H}} &= \langle \boldsymbol{f}, \mathbf{G}(\cdot, x) \rangle_{\mathcal{H}} = \langle \mathbf{P}\boldsymbol{f}, \mathbf{PG}(\cdot, x) \rangle_{\mathbb{R}^n} = \langle \boldsymbol{f}, \mathbf{P}^\top\mathbf{PG}(\cdot, x) \rangle_{\mathbb{R}^n} \\ &= \langle \boldsymbol{f}, \delta(\cdot, x) \rangle = f(x), \end{aligned}$$

where $\mathbf{G}(\cdot, x)$ is the column of $\mathbf{G}$. Every linear functional on a finite dimensional vector space is continuous, and so is the Dirac's. Therefore, $\mathcal{H}$ is an RKHS determined by the kernel $G$. □

## 5.2 Explicit ODEs

We consider dynamical systems with $d$ interacting elements evolving in some closed time interval $[0, T]$. We denote by $x_k : [0, T] \to \mathbb{R}$ for $k = 1, \ldots, d$, the functions describing the evolution of the elements of the system and by $u_k : [0, T] \to \mathbb{R}$ for $k = 1, \ldots, m$, the action of $m$ external forces. In compact notation, we denote by $\boldsymbol{x}(t) = (x_1(t), \ldots, x_d(t))^\top$, $\boldsymbol{u}(t) = (u_1(t), \ldots, u_m(t))^\top$ the vectors of state variables and external forces respectively. We assume that each state variable $x_k$ satisfies

$$P_{\theta_k} x_k(t) = f_k(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\beta}), \;\; k = 1, \ldots, d, \;\; t \in [0, T],$$

where $P_{\theta_k} = d/dt + \theta_k \mathrm{I}$ is the linear differential operator associated with the $k$th equation of the system. Here, $f_k$ is a known function depending on $t$ through $\boldsymbol{x}(t)$ and $\boldsymbol{u}(t)$, and $\boldsymbol{\beta}$ is a vector of parameters. We use notation I for the identity operator. For differential operators of a higher order see González et al. (2014).

In the sequel, we refer to the whole set of the parameters of the system by $\{\boldsymbol{\theta}, \boldsymbol{\beta}\}$, where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_d)^\top$. Typically, a noisy sample of $\boldsymbol{x}$ is observed at a grid of time points $\boldsymbol{t} = (t_1, \ldots, t_n)^\top$. Let $\boldsymbol{y}_k$ indicate the available data for state $k$ and let $x_k(\boldsymbol{t})$ indicate the vector of values corresponding to the evaluated $k$th state at time points $\boldsymbol{t}$, i.e.

$$\boldsymbol{y}_k = y_k(\boldsymbol{t}) = (y_k(t_1), \ldots, y_k(t_n))^\top, \tag{5.5}$$

$$\boldsymbol{x}_k = x_k(\boldsymbol{t}) = (x_k(t_1), \ldots, x_k(t_n))^\top, \tag{5.6}$$

$$\boldsymbol{f}_k = (f_k(\boldsymbol{x}(t_1), \boldsymbol{u}(t_1), \boldsymbol{\beta}), \ldots, f_k(\boldsymbol{x}(t_n), \boldsymbol{u}(t_n), \boldsymbol{\beta}))^\top. \tag{5.7}$$

We consider a model with additive noise

$$y_k(\boldsymbol{t}) = x_k(\boldsymbol{t}) + \varepsilon_k(\boldsymbol{t}),$$

where $\epsilon_k$ represents a noise process for the $k$th state. We assume that $\boldsymbol{\varepsilon}_k(\boldsymbol{t})$ is an independent multivariate zero-mean Gaussian noise with variance $\sigma_k^2$ for each $k$th state. Let $\boldsymbol{y}(\boldsymbol{t})$ and $\boldsymbol{x}(\boldsymbol{t})$ denote the matrices that comprise all the rows $\boldsymbol{y}_k$, $\boldsymbol{x}_k$, for $k = 1, \ldots, d$, respectively. Then the *log-likelihood* of the model given $\boldsymbol{y}(\boldsymbol{t})$ is

$$l(\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{x}(\boldsymbol{t}) | \boldsymbol{y}(\boldsymbol{t})) = -\sum_{k=1}^{d} \frac{1}{2\sigma_k^2} \|\boldsymbol{y}_k - \boldsymbol{x}_k\|^2 \;\text{ where }\; P_{\theta_k} x_k(t) = f_k(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\beta}). \tag{5.8}$$

Here, $\boldsymbol{x}$ depends on $\{\boldsymbol{\theta}, \boldsymbol{\beta}\}$ although for the sake of simplicity it is not explicitly specified. Since ODEs generally do not have a closed-form solution, evaluating the log-likelihood at a specific value of the parameters $\boldsymbol{\theta}, \boldsymbol{\beta}$ requires solving numerically the system for those parameters. Thus, the numerical maximization of the log-likelihood function involves repeatedly solving the system for every new iteration in the optimization algorithm. This approach is thus computationally costly, especially for large systems.

## 5.3 RKHS based penalized log-likelihood

In this section, our goal is to reformulate (5.8) in terms of a penalized loss so as to obtain a computationally tractable solution that does not require an explicit solution of the system of ODEs. To this aim, we discretize the system and consider the difference equation

$$\mathbf{P}_{\theta_k} \boldsymbol{x}_k = \boldsymbol{f}_k,$$

where $\boldsymbol{x}_k$ and $\boldsymbol{f}_k$ are defined in (5.6) and (5.7) and $\mathbf{P}_{\theta_k}$ is the operator defined by

$$\mathbf{P}_{\theta_k} = \mathbf{D}_n + \theta_k \mathbf{I}_n. \tag{5.9}$$

Here, $\mathbf{I}_n$ is the identity matrix of order $n$ and $\mathbf{D}_n \in \mathbb{R}^{n \times n}$ is the first order difference matrix

$$\mathbf{D}_n = \Delta^{-1} \begin{pmatrix} -1 & 1 & & & & \\ -1 & 0 & 1 & & & \\ & & \ddots & & & \\ & & -1 & 0 & 1 & \\ & & & -1 & 1 & \end{pmatrix}, \tag{5.10}$$

where $\Delta = \operatorname{diag}(t_2 - t_1, 2(t_3 - t_1), \ldots, 2(t_{n-1} - t_{n-2}), t_n - t_{n-1})$. The difference in finite approximations used for $t_1$ and $t_n$ is due to the fact that they are boundary points. We aim to define a penalized loss function

$$l_\lambda(\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{x}(\boldsymbol{t}) | \boldsymbol{y}(\boldsymbol{t})) = -\sum_{k=1}^{d} \frac{1}{2\sigma_k^2} \|\boldsymbol{y}_k - \boldsymbol{x}_k\|^2 - \frac{\lambda}{2} \sum_{k=1}^{d} \Omega(\boldsymbol{x}_k), \tag{5.11}$$

where $\Omega$ defines a norm of $\boldsymbol{x}_k$ in an RKHS.

First consider the homogeneous system, i.e. $\boldsymbol{f}_k = \mathbf{0}_n$. In this case, according to

theorem 5.3 defining the penalty as $\Omega(\boldsymbol{x}_k) = \|\mathbf{P}_{\theta_k}\boldsymbol{x}_k\|^2$ implies that

$$\Omega(\boldsymbol{x}_k) = \|\boldsymbol{x}_k\|^2_{\mathcal{H}_{\theta_k}},$$

where $\mathcal{H}_{\theta_k}$ is the RKHS defined by the kernel matrix

$$\mathbf{K}_{\theta_k} = (\mathbf{P}^\top_{\theta_k}\mathbf{P}_{\theta_k})^{-1}. \tag{5.12}$$

Consequently, according to the representer theorem 5.2 the penalized loss has the minimizer $\widehat{\boldsymbol{x}}_k$ of the form

$$\widehat{\boldsymbol{x}}_k = \mathbf{K}_{\theta_k}\widehat{\boldsymbol{\alpha}}_k,$$

where $\boldsymbol{\alpha}_k = (\mathbf{K}_{\theta_k} + \lambda\sigma_k^2\mathbf{I}_n)^{-1}\boldsymbol{y}_k$.

In general, the interest in inferring parameters of ODEs is for nonhomogeneous systems. In the same spirit as above, one might consider

$$\Omega(\boldsymbol{x}_k) = \|\mathbf{P}_{\theta_k}\boldsymbol{x}_k - \boldsymbol{f}_k\|^2, \tag{5.13}$$

as a penalty. However, (5.13) cannot be used directly as a penalty for two reasons. Expression (5.13) cannot be reformulated as a norm of $\boldsymbol{x}_k$ in an RKHS; when $\boldsymbol{x}_k = \mathbf{0}_n$ then $\|\mathbf{P}_{\theta_k}\boldsymbol{x}_k - \boldsymbol{f}_k\|^2$ is not necessarily zero. Also, in this case the equations of the system are not independent. In a general setting, each $\boldsymbol{x}_k$ is affected by $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$.

To circumvent the previous problem we consider that each $f_k$ is a function of $\boldsymbol{\beta}$ that depends on $t$ through $\boldsymbol{u}(t)$ and some fixed surrogate of $\boldsymbol{x}$, denoted by $\boldsymbol{x}^\star(t)$, independent of $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$. This step represents a linearization of the system and makes the equations independent of each other. In subsection 4 we elaborate on the definition of an appropriate $\boldsymbol{x}^\star(t)$.

In order to find an RKHS representation of the system in a general case, as before, we assume that $\mathbf{P}_{\theta_k}$ is invertible for each $k$ . Denote by

$$\begin{aligned}\boldsymbol{f}_k^\star &= (f_k(\boldsymbol{x}^\star(t_1), \boldsymbol{u}(t_1), \boldsymbol{\beta}), \ldots, f_k(\boldsymbol{x}^\star(t_n), \boldsymbol{u}(t_n), \boldsymbol{\beta}))^\top, \\ \widetilde{\boldsymbol{x}}_k &= \boldsymbol{x}_k - \mathbf{P}^{-1}_{\theta_k}\boldsymbol{f}_k^\star.\end{aligned} \tag{5.14}$$

Since $\mathbf{P}_{\theta_k}$ is a linear operator we obtain that for all $k = 1, \ldots, d$

$$\|\mathbf{P}_{\theta_k}\boldsymbol{x}_k - \boldsymbol{f}_k^\star\|^2 = \|\mathbf{P}_{\theta_k}\widetilde{\boldsymbol{x}}_k\|^2,$$

which according to theorem 5.3 is a norm of $\widetilde{\boldsymbol{x}}_k$ in an RKHS defined by the kernel (5.12). Hence, we write

$$\Omega(\widetilde{\boldsymbol{x}}_k) = \|\mathbf{P}_{\theta_k}\widetilde{\boldsymbol{x}}_k\|^2 = \|\widetilde{\boldsymbol{x}}_k\|^2_{\mathcal{H}_{\theta_k}}.$$

In practice, noise samples are obtained for $\boldsymbol{x}_k$ and not for $\widetilde{\boldsymbol{x}}_k$. Therefore, the inference problem on $\widetilde{\boldsymbol{x}}_k$ requires the following transformation of the original data

$$\widetilde{\boldsymbol{y}}_k = \boldsymbol{y}_k - \mathbf{P}^{-1}_{\theta_k}\boldsymbol{f}^\star_k.$$

It is straightforward to check that $\widetilde{\boldsymbol{y}_k} \sim \mathcal{N}(\widetilde{\boldsymbol{x}}_k(\boldsymbol{t}), \sigma_k^2\mathbf{I}_n)$ and therefore the variance of the $y_{kj}$'s is the same as the variance of the $\widetilde{y}_{kj}$'s.

## 5.4 Approximate ODEs inference

The goal of this section is to provide computational details to infer the set of parameters $\{\boldsymbol{\theta}, \boldsymbol{\beta}\}$ by using the approximate ODEs representation described in Section 5.3. As detailed there, a definition of each $\boldsymbol{x}^\star_k$, a surrogate of $\boldsymbol{x}_k$ is required. In this thesis, we express each $\boldsymbol{x}^\star_k$ in terms of a spline basis expansion,

$$\boldsymbol{x}^\star_k = \sum_{i=1}^{q} c_{ki}\phi_i(\boldsymbol{t}), \tag{5.15}$$

where $q$ is the number of basis functions $\phi_i$ and $c_{ki}$, $i = 1, \ldots, 1$ are coefficients of the expansion. The number of basis functions should be large enough to capture the variation in the solutions of the system of ODEs .

**Definition 5.5.** *Let $\boldsymbol{y}_k$, $\boldsymbol{x}_k$, $\mathbf{P}_{\theta_k}$, $\boldsymbol{f}^\star_k$ and $\boldsymbol{x}^\star_k$ be defined by* (5.5), (5.6), (5.9), (5.14) *and* (5.15) *for $k = 1, \ldots, d$. We define the approximated penalized pseudo log-likelihood of the ODEs model associated to (5.11) as*

$$l_\lambda(\boldsymbol{\theta}, \boldsymbol{\beta}|\boldsymbol{y}(\boldsymbol{t}), \boldsymbol{x}^\star(\boldsymbol{t})) = -\sum_{k=1}^{d}\frac{1}{2\sigma_k^2}\|\boldsymbol{y}_k - \boldsymbol{x}_k\|^2 - \frac{\lambda}{2}\sum_{k=1}^{d}\|\mathbf{P}_{\theta_k}\boldsymbol{x}_k - \boldsymbol{f}^\star_k\|^2,$$

*where $\lambda > 0$ and $\boldsymbol{x}^\star(\boldsymbol{t}) = (x^\star_1(\boldsymbol{t}), \ldots, x^\star_d(\boldsymbol{t}))^\top$.*

Since for all $k = 1, \ldots, d$ it holds

$$\|\boldsymbol{y}_k - \boldsymbol{x}_k\|^2 \;=\; \|\boldsymbol{y}_k - \mathbf{P}^{-1}_{\theta_k}\boldsymbol{f}^\star_k - (\boldsymbol{x}_k - \mathbf{P}^{-1}_{\theta_k}\boldsymbol{f}^\star_k)\|^2 = \|\widetilde{\boldsymbol{y}}_k - \widetilde{\boldsymbol{x}}_k\|^2,$$

it follows that we can rewrite $l_\lambda(\boldsymbol{\theta}, \boldsymbol{\beta}|\boldsymbol{y}(\boldsymbol{t}), \boldsymbol{x}^\star(\boldsymbol{t}))$ as follows

$$l_\lambda(\boldsymbol{\theta}, \boldsymbol{\beta}|\boldsymbol{y}(\boldsymbol{t}), \boldsymbol{x}^\star(\boldsymbol{t})) = -\sum_{k=1}^{d} \frac{1}{2\sigma_k^2} \|\widetilde{\boldsymbol{y}}_k - \widetilde{\boldsymbol{x}}_k\|^2 - \frac{\lambda}{2} \sum_{k=1}^{d} \|\mathbf{P}_{\theta_k} \widetilde{\boldsymbol{x}}_k\|^2. \tag{5.16}$$

This form allows us to use the representer theorem according to which the minimizer of (5.16) is given by $\widehat{\widetilde{\boldsymbol{x}}}_k = \mathbf{K}_{\theta_k} \widehat{\boldsymbol{\alpha}}_k$, where $\widehat{\boldsymbol{\alpha}}_k = (\mathbf{K}_{\theta_k} + \lambda \sigma_k^2 \mathbf{I}_n)^{-1} \widetilde{\boldsymbol{y}}_k$. It follows that

$$\widehat{\widetilde{\boldsymbol{x}}} = \mathbf{S}_{\lambda,k} \widetilde{\boldsymbol{y}}_k, \tag{5.17}$$

where

$$\mathbf{S}_{\lambda,k} = \mathbf{K}_{\theta_k} (\mathbf{K}_{\theta_k} + \lambda \sigma_k^2 \mathbf{I}_n)^{-1} \widetilde{\boldsymbol{y}}_k. \tag{5.18}$$

Hence, the minimizer written in terms of $\boldsymbol{x}_k$ is given by

$$\widehat{\boldsymbol{x}}_k = \mathbf{K}_{\theta_k} \widehat{\boldsymbol{\alpha}}_k + \mathbf{P}_{\theta_k}^{-1} \boldsymbol{f}_k^\star.$$

Next, we show how to compute $l_\lambda$ in practice.

**Proposition 5.1.** *Assume that* $\mathbf{P}_{\theta_k}^{-1}$ *exists. Then it holds*

$$\{\widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{\beta}}\} = arg \max_{\boldsymbol{\theta}, \boldsymbol{\beta}} l_\lambda(\boldsymbol{\theta}, \boldsymbol{\beta}|\boldsymbol{y}(\boldsymbol{t}), \boldsymbol{x}^\star(\boldsymbol{t})) = arg \max_{\boldsymbol{\theta}, \boldsymbol{\beta}} g_\lambda(\boldsymbol{\theta}, \boldsymbol{\beta}|\boldsymbol{y}(\boldsymbol{t}), \boldsymbol{x}^\star(\boldsymbol{t})),$$

where

$$g_\lambda(\boldsymbol{\theta}, \boldsymbol{\beta}|\boldsymbol{y}(\boldsymbol{t}), \boldsymbol{x}^\star(\boldsymbol{t})) = -\sum_{k=1}^{d} \frac{1}{2\sigma_k^2} \widetilde{\boldsymbol{y}}_k^\top \left\{ \mathbf{I}_n - (\mathbf{I}_n + \sigma_k^2 \lambda \mathbf{K}_{\theta_k}^{-1})^{-1} \right\} \widetilde{\boldsymbol{y}}_k. \tag{5.19}$$

Optimization of (5.19) with a conjugate gradient method produces estimates of $\{\boldsymbol{\theta}, \boldsymbol{\beta}\}$. If the set of parameters of the system is separable by equations, independent optimization for those can be done, thus helping to avoid local maxima and speed up the procedure. Finally, estimates for each $\widehat{\boldsymbol{x}}_k$ are available by means of

$$\widehat{\boldsymbol{x}}_k = \mathbf{K}_{\widehat{\theta}_k} \widehat{\boldsymbol{\alpha}}_k + \mathbf{P}_{\widehat{\theta}_k}^{-1} f_k(\boldsymbol{u}(\boldsymbol{t}), \boldsymbol{x}^\star(\boldsymbol{t}), \widehat{\boldsymbol{\beta}}),$$

where $\widehat{\boldsymbol{\alpha}}_k = (\mathbf{K}_{\widehat{\theta}_k} + \lambda \sigma_k^2 \mathbf{I}_n)^{-1} \widetilde{\boldsymbol{y}}_k$.

### 5.4.1 Model selection

In our penalized approach, the value of the nuisance parameter $\lambda$ has to be fixed. For this purpose, we derive the AIC using results from Hastie et al. (2009). For the linear smoother (5.17) of the $k$th state the influence matrix is given by (5.18). The effective number of parameters for each state is defined as $\mathrm{df}_k = \mathrm{tr}(\widehat{\mathbf{S}}_{\lambda,k}) = \mathrm{tr}\{\mathbf{K}_{\widehat{\theta}_k}(\mathbf{K}_{\widehat{\theta}_k} + \lambda\sigma_k^2\mathbf{I}_n)\}^{-1}$. Then the AIC is defined in our context as

$$\mathrm{AIC}(\lambda) = -2l_\lambda(\boldsymbol{\theta}, \boldsymbol{\beta}|\boldsymbol{y}(\boldsymbol{t}), \boldsymbol{x}^\star(\boldsymbol{t})) + 2\sum_{k=1}^{d}\mathrm{df}_k.$$

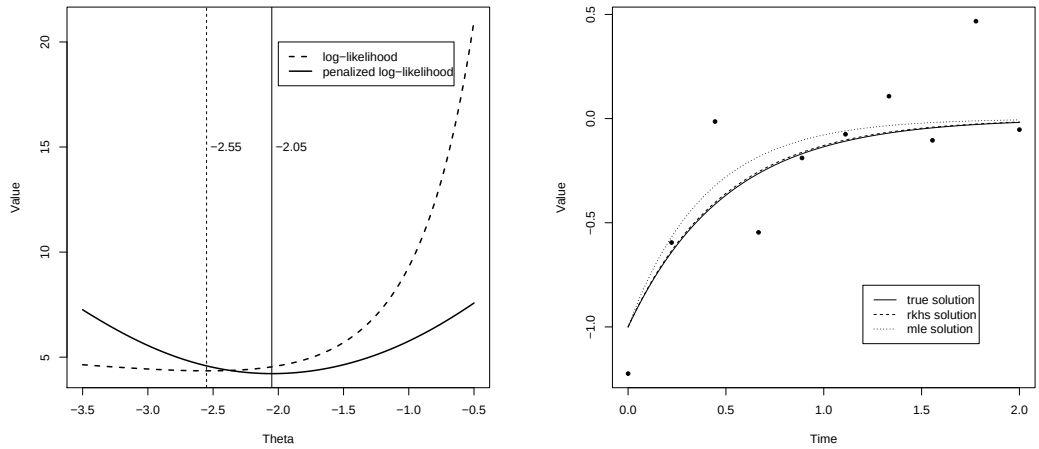
## 5.5 Examples using synthetically generated data

### 5.5.1 Explicit ODEs versus regularization approach

In this section we use a toy example to illustrate the advantages of using a regularization approach to estimate the parameters of a dynamical system. We consider the differential equation

$$x'(t) = \theta x(t).$$

For fixed $\theta$ and initial condition $x(0)$, the solution of the differential equation is given by $x(t) = x(0)\exp(\theta t)$. We fix $\theta = -2$, $x(0) = -1$ and we generate 500 samples of 10 equally spaced points in the interval $[0, 2]$ using Gaussian noise with $\sigma = 0.25$. For each sample we calculate the maximum likelihood estimator (MLE) of $\theta$ and our RKHS based estimator for $\lambda$ selected by means of the AIC. The average absolute deviance to the true parameter of the MLE is 0.73 with a standard deviation of 1.03 whereas the average error for the penalized approach is 0.53 with a standard deviation of 0.38. In Figure 5.1 we show the results for one run of the experiment. Figure 5.1 a) shows the negative log-likelihood and the penalized log-likelihood of the model for one of the generated data sets. The penalized approach results in an "improvement" of the original log-likelihood for the parameter estimation with the minimum closer to the true value of the parameter. Also, the original negative log-likelihood becomes extremely flat for small values of the parameters, which can produce computational problems in the optimization step. Figure 5.1 b) shows the MLE and RKHS solutions together with the true function $x(t) = -\exp(-2t)$ for $t \in [0, 2]$. In this example, penalizing the log-likelihood improves the estimator. The true function $x$ is better approximated using the penalized approach due to the finite sample bias of the MLE.




(a) Negative log-likelihood and the penalized log-likelihood of the model for one of the generated data sets.

(b) Simulated data, true solution, and the two estimated solutions using the MLE and RKHS approaches.

Fig. 5.1 The results obtained for the differential equation $x'(t) = \theta x(t)$.

Also, the estimate of the parameter is closer to the true value of $\theta$ in this particular realization ($\widehat{\theta}_{MLE} = -2.55$ vs. $\widehat{\theta}_{RKHS} = -2.05$).

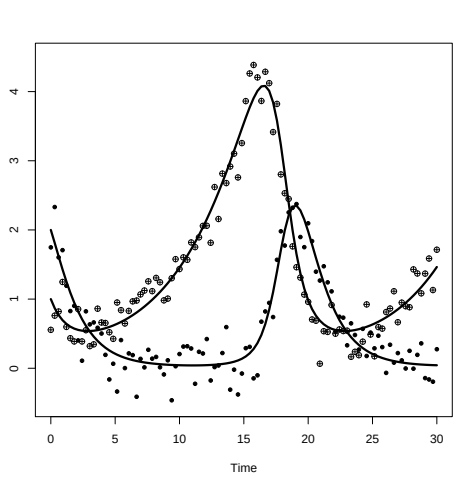
### 5.5.2 Comparison with the MLE

In this section, we work with the Lotka-Volterra system of differential equations originally proposed in the theory of auto-catalytic chemical reaction (Lotka, 1910). The system is

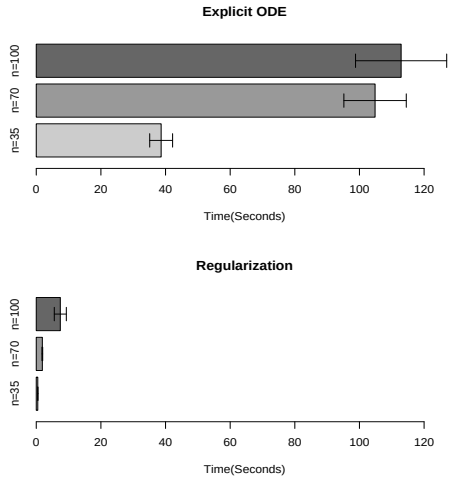
$$x_1'(t) = x_1(t)\{\theta_1 - \beta_1 x_2(t)\},$$
$$x_2'(t) = -x_2(t)\{\theta_1 - \beta_2 x_2(t)\},$$

where $\boldsymbol{\theta} = (\theta_1, \theta_2)^\top$ and $\boldsymbol{\beta} = (\beta_1, \beta_2)^\top$ are the parameters.

Our aim is to evaluate the accuracy and speed of our RKHS penalized approach in comparison with the classical MLE approach. To do so, we run a simulation study for fixed $\theta_1 = 0.2$, $\beta_1 = 0.35$, $\theta_2 = 0.7$, $\beta_2 = 0.40$ and initial conditions $x_1(0) = 1$ and $x_2(0) = 2$. We generate samples made up of $n$ equally spaced independent observations of the state variables $x_1$ and $x_2$ in the interval $[0, 30]$ that we perturb with zero mean Gaussian noise with standard deviation $\sigma$. We generate data for the sample sizes $n = 35, 70, 100$ and two noise scenarios $\sigma = 0.1, 0.25$. In Figure 5.2 a) we show the true solutions of the model for the above mentioned parameters together with the data




(a) True solutions and the data in the Lotka-Volterra experiment for $n = 100$ and $\sigma = 0.25$.




(b) Time comparison between the explicit MLE approach and the proposed penalized RKHS based approach.

Fig. 5.2 The results obtained for the Lotka-Volterra equations.

of one of the simulations.

In order to apply the proposed approach we obtain the functions $x_1^\star$ and $x_2^\star$ using penalized splines and $\lambda = 100$. To perform the MLE estimation we use 10 different initial values of the parameters, randomly generated in the interval $[0, 1]$, and we use the log-likelihood value to select the best candidate.

In Figure 5.2 b) we show a comparison of average running times. The RKHS-based estimator is 120.08, 24.06 and 14.41 times faster than the explicit ODE approach for $n = 35, 70$ and 100, respectively. In Table 5.1, we show the mean square errors of the estimates with respect to the true parameters for 100 runs of the experiment. For $n = 35$ the penalized RKHS approach performs significantly better than the explicit ODE estimates, which is explained by the empirical bias suffered by the MLE approach illustrated in Section 5.5.1. For $n = 75, 100$ both methods are similar in terms of precision. The noise in the data is reflected in the precision of the estimates for both techniques; in all cases the errors are larger for $\sigma = 0.25$.

### 5.5.3 Influence of the sample size on the estimation

Since we were not able to prove consistency of the proposed estimator, in this section we test the proposed methodology in order to see the influence of the sample size on the estimation of the parameters. We consider the FitzHugh-Nagumo model (FHN)

| Lotka-Volterra ODE model | | | | | | |
|---|---|---|---|---|---|---|
| $\sigma$ | n | Method | $\lvert\theta_1 - \widehat{\theta}_1\rvert$ | $\lvert\beta_1 - \widehat{\beta}_1\rvert$ | $\lvert\theta_2 - \widehat{\theta}_2\rvert$ | $\lvert\beta_2 - \widehat{\beta}_2\rvert$ |
| 0.1 | 35 | RKHS | **0.0002 (0.0003)** | **0.0007 (0.0007)** | **0.0031 (0.0036)** | **0.0014 (0.0014)** |
| | | MLE | 0.0016 (0.0088) | 0.0063 (0.0425) | 0.0422 (0.1809) | 0.0227 (0.1064) |
| | 70 | RKHS | 0.0001 (0.0001) | 0.0002 (0.0002) | **0.0009 (0.0011)** | **0.0003 (0.0004)** |
| | | MLE | **0.0000 (0.0001)** | **0.0001 (0.0006)** | 0.0017 (0.0034) | 0.0005 (0.0010) |
| | 100 | RKHS | 0.0001 (0.0001) | **0.0001 (0.0002)** | **0.0005 (0.0006)** | **0.0002 (0.0002)** |
| | | MLE | **0.0000 (0.0001)** | 0.0002 (0.0010) | 0.0013 (0.0023) | 0.0004 (0.0008) |
| 0.25 | 35 | RKHS | **0.0010 (0.0013)** | **0.0017 (0.0024)** | **0.0111 (0.0205)** | **0.0038 (0.0059)** |
| | | MLE | 0.0029 (0.0180) | 0.0081 (0.0392) | 0.0173 (0.0487) | 0.0078 (0.0359) |
| | 70 | RKHS | **0.0004 (0.0006)** | **0.0008 (0.0009)** | **0.0042 (0.0047)** | **0.0015 (0.0019)** |
| | | MLE | 0.0007 (0.0025) | 0.0030 (0.0115) | 0.0151 (0.0474) | 0.0062 (0.0301) |
| | 100 | RKHS | **0.0003 (0.0004)** | **0.0005 (0.0006)** | **0.0034 (0.0043)** | **0.0011 (0.0016)** |
| | | MLE | 0.0008 (0.0032) | 0.0028 (0.0116) | 0.0174 (0.0603) | 0.0083 (0.0387) |

Table 5.1 Mean square error for the inferred parameters in the Lotka-Volterra model. Standard deviations shown in parenthesis. The true value of the parameters are fixed to $\theta_1 = 0.2$, $\beta_1 = 0.35$ $\theta_2 = 0.7$, $\beta_2 = 0.40$. The best result for each comparison is boldfaced.

(FitzHugh, 1955) which is described by the equations

$$\begin{aligned} x_1'(t) &= c\{x_1(t) - \tfrac{x_1(t)^3}{3} + x_2(t)\}, \\ x_2'(t) &= \tfrac{1}{c}\{x_1(t) - a + bx_2(t)\}, \end{aligned} \tag{5.20}$$

where $a$, $b$ and $c$ are the parameters of the system and $x_1$ and $x_2$ are the state variables.

In our experiment, we fix the parameters $a, b = 0.2$ and $c = 3$ and initial conditions $x_{1,0} = -1$ and $x_{2,0} = -1$. We generate samples $\mathcal{D} = \{(y_{ki}, t_i) \in \mathbb{R} \times [0, 20]\}_{i=1}^n$ for $i = 1, 2$ made up of independent noisy observations of the state variables $x_1$ and $x_2$ at $n$ fixed (and equally spaced) time points $t_1, \ldots, t_n$ in the interval $[0, 20]$. We use a normal scheme $y_{ki} \sim \mathcal{N}(x_k(t_i), \sigma^2)$, where the variance $\sigma^2$ is assumed equal for $x_1$ and $x_2$ with a value of 0.1. Within this framework, we estimate the parameters of the FHN model for sample sizes $n$ ranging between 30 and 200. In Figure 5.3, we show the true functions $x_1$ and $x_2$ and the data generated for $n = 100$.

In order to apply the proposed methodology we need to make the system homogeneous as detailed in Section 5.3. Since $x_1$ is non-linearly included in (5.20) we fit spline functions to the data. Then we replace $x_1$ and $x_2$ by their spline estimates $x_1^\star$

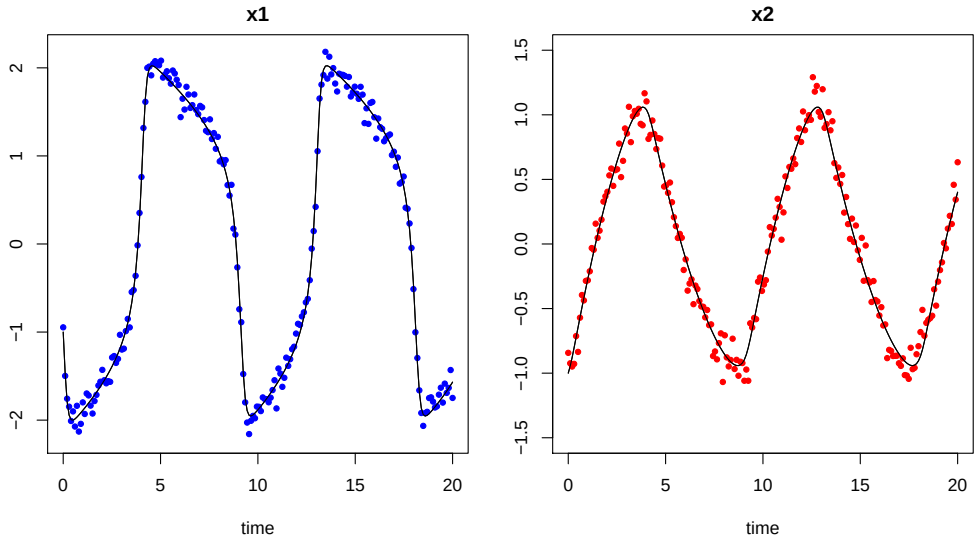



Fig. 5.3 Solution of the FHN model and generated data points in one run of the experiment ($n = 100$).

and $x_2^\star$ which yields:

$$x_1'(t) - cx_1(t) \approx c\{-\tfrac{x_1^{\star 3}(t)}{3} + x_2^\star(t)\},$$
$$x_2'(t) - \tfrac{b}{c}x_2(t) \approx \tfrac{1}{c}\{x_1^\star(t) - a\}.$$

Under the previous transformation, the right part of the system does not depend on $x_1$ and $x_2$ and we can proceed as explained in Section 5.3. Consider $\mathbf{D}_n$, the difference matrix defined in (5.10) for a particular sample size $n$. Then the difference matrix of the system is given by

$$\mathbf{P}_{b,c} = \begin{pmatrix} \mathbf{D}_n - c\mathbf{I}_n & \mathbf{O}_n \\ \mathbf{O}_n & \mathbf{D}_n - \frac{b}{c}\mathbf{I}_n \end{pmatrix},$$

where $\mathbf{I}_n$ and $\mathbf{O}_n$ are, respectively, the $n$-dimensional identity matrix and $n$-dimensional zero matrix. Under the previous formulation, the kernel matrix used to penalize the log-likelihood can be explicitly written as $\mathbf{K}_{b,c} = (\mathbf{P}_{b,c}^\top \mathbf{P}_{b,c})^{-1}$, which connects the system of differential equations with the RKHS framework.

We apply the proposed methodology to estimate the parameters of the FHN system. For each sample size we generate 100 independent samples and we estimate the parameters $a$, $b$, $c$ and $\sigma^2$. Following the discussion in Section 5.4.1 the regularization parameter $\lambda$ was fixed to 10000. In this case, the variance $\sigma^2$ is estimated as

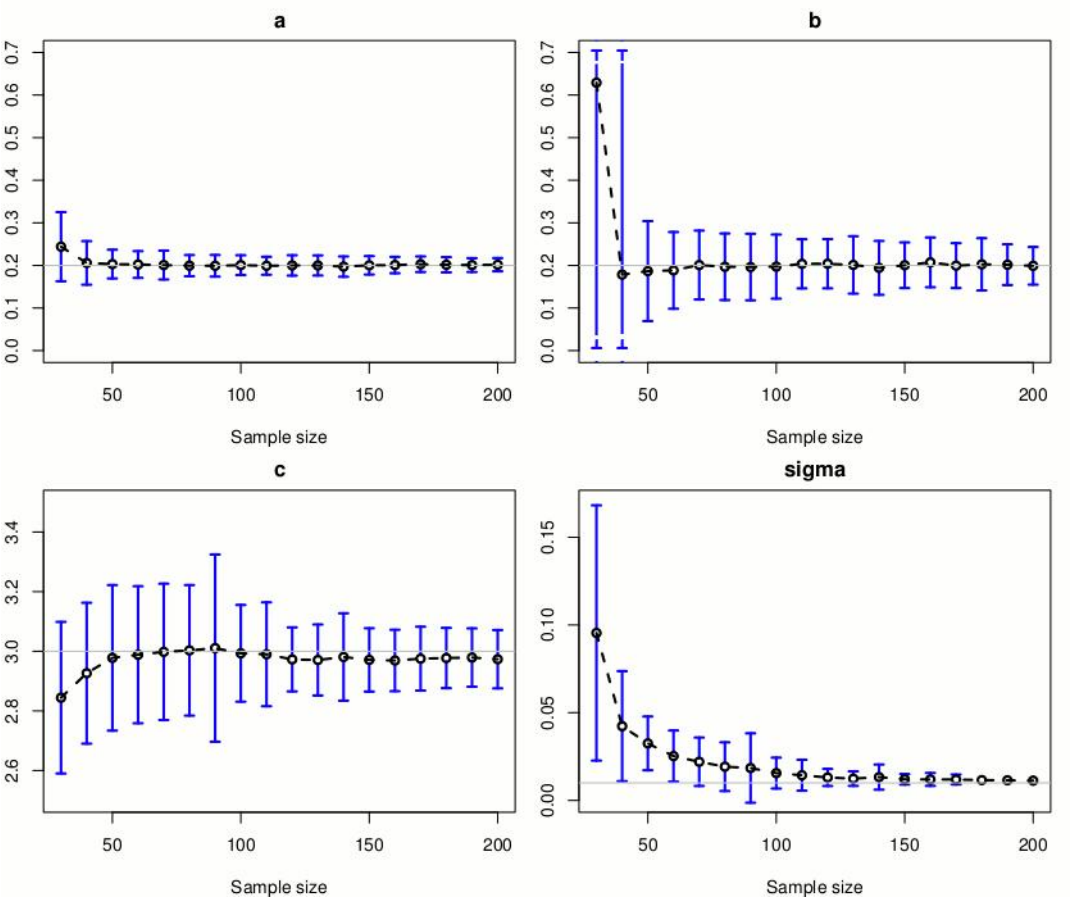



Fig. 5.4 95% confidence intervals for the parameters $a$, $b$, $c$ and $\sigma^2$ of the FHN equations for different sample sizes. Horizontal grey lines represent the true values of the parameters. The results are obtained using 50 runs of the experiment.

a parameter of the model. In Figure 5.4 we show the 95% confidence intervals for the four parameters across different sample sizes. When $n$ increases the estimation of the parameters is more precise, but the trend indicates that the method is either not consistent or requires a lot of data.

### 5.5.4 Comparison with generalized profiling procedure

To conclude this section we compare the results obtained in the FHN system with those obtained by the *generalized profiling* approach proposed in Ramsay et al. (2007). In particular, we compare the estimates for two scenarios with sample sizes $n = 50$ and $n = 300$. We run 50 replicates of the experiment and show the results in Table 5.2. We compare both methods in terms of the differences between the estimates of the

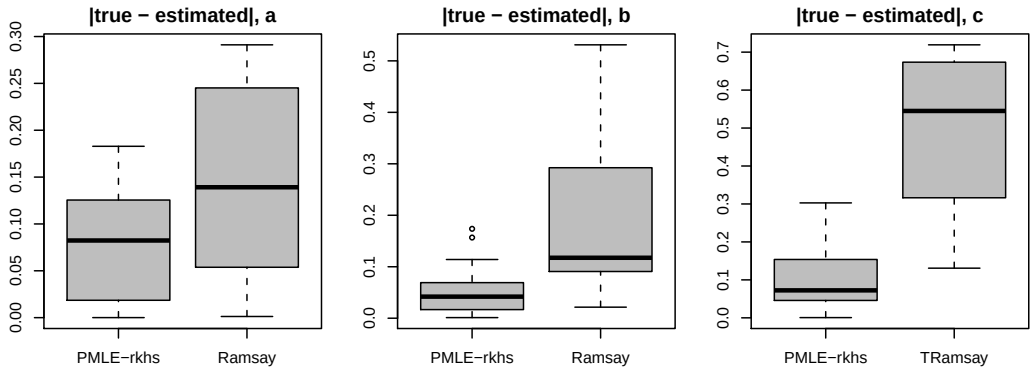
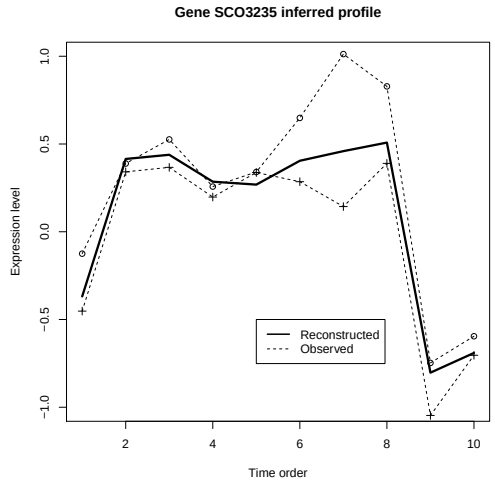



Fig. 5.5 Box-plots for the absolute errors $|\widehat{a}_i - a|$, $|\widehat{b}_i - b|$ and $|\widehat{c}_i - c|$ in the estimation of the parameters of the FHN equations. The results are obtained using the generalized profiling and the maximum penalized likelihood approach (MPLE-rkhs) proposed in this work for $n = 50$.

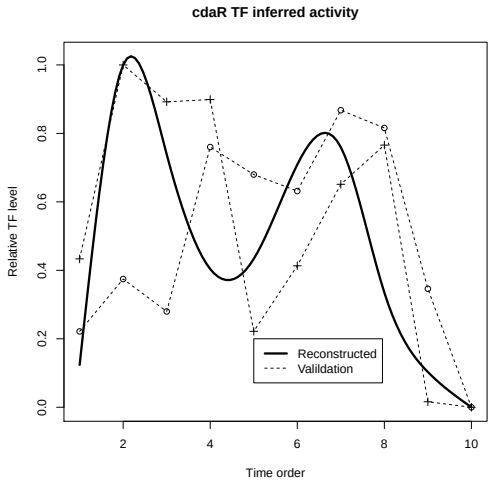
| | FHN | Av. error | | Max. error | | Min. error | |
|---|---|---|---|---|---|---|---|
| | params. | MPLE | Ramsay | MPLE | Ramsay | MPLE | Ramsay |
| n = 50 | a | **0.0780** | 0.2892 | **0.1829** | 0.5130 | **0.0000** | 0.0019 |
| | b | **0.0485** | 0.7023 | **0.1737** | 1.2936 | **0.0012** | 0.0619 |
| | c | **0.0983** | 0.2503 | **0.3028** | 0.6253 | **0.0006** | 0.0488 |
| n = 300 | a | 0.0058 | **0.0032** | 0.0150 | **0.0126** | 0.0000 | 0.0000 |
| | b | 0.0175 | **0.0101** | 0.0609 | **0.0318** | 0.0005 | **0.0002** |
| | c | 0.0348 | **0.0134** | 0.1133 | **0.0476** | 0.0008 | **0.0010** |

Table 5.2 Average, maximum and minimum errors for the estimation of the parameters of the FHN system achieved by the generalized profiling and the maximum penalized likelihood approach (MPLE-RKHS) proposed in this work. Two different sample sizes $(50, 300)$ are used for the comparison. The best result for each comparison is boldfaced.

parameters and their true values. The average, maximum and minimum errors across the 50 replicates are used to compare both approaches. For $n = 50$ the proposed methodology improves the Ramsay et al. (2007) method for the three parameters irrespective of the criteria. Box-plots of the error distributions are shown in Figure 5.5. For $n = 300$ only minor differences have been found.




(a) Gene SCO3235, reconstructed profile. Circles and crosses represent the observed data and lines the obtained profiles. The estimated variance is $\widehat{\sigma}^2 = 0.014$. The estimated parameters for this gene are $\beta_1 = 0.65$ (stdev= 0.57), $\beta_2 = 1.07(0.51)$, $\beta_3 = 2.09$ (0.26) and $\theta$=1.06 (0.21).




(b) Reconstructed activity of the master activator cdaR scaled between 0 and 1. Circles and crosses represent the data obtained in two independent experiments not used in the estimation process.

Fig. 5.6 Reconstructed genes profiles and master activator cdaR.

## 5.6 Real example: Reconstruction of Transcription Factor activities in Streptomyces coelicolor

A gene regulatory network consists of a gene encoding a transcription factor (TF); together with the genes it regulates. In the absence of reliable technology to measure the activity of the TF (number of TF-protein molecules in the cell), the problem is to reconstruct it from the gene expression data of its target genes.

In this section, we work with a data set of gene expression levels in the *Streptomyces coelicolor* bacterium. The goal is to reconstruct the activity of the transcription factor (TF) *CdaR* using 10-point time-series gene expression data of 17 genes. For each gene, two different series corresponding to a wild type and a mutant type bacterium (for which a transcriptional regulator cdaR has been knocked out) are available. Measurements are available at time points (in mins.) $\boldsymbol{t} = \{16, 18, 20, 21, 22, 23, 24, 25, 39, 67\}$. The importance of understanding the behaviour of the cdaR is that it is partially responsible for the production of a particular type of antibiotic.

Following Khanin et al. (2007) we assume that changes in the expression levels of

the genes are caused by changes in the cdaR protein and the mRNA degradation. We denote by $\eta(t)$ the activity profile of the regulator cdaR at time $t$, and by $x_k(t)$ the expression level of each gene $k$ at time $t$. This regulatory system is modelled by

$$x_k'(t) + \theta_k x_k(t) = \beta_{1i} + \beta_{2i} \frac{\eta(t)}{\beta_{3i} + \eta(t)}, \tag{5.21}$$

where $\theta_k$ is the rate of mRNA degradation, $\beta_{2i}$ and $\beta_{3i}$ are gene-specific kinetic parameters for the gene $k$, and $\beta_{1i}$ is an additive constant that accounts for the basal level of transcription and the nuisance effects from micro-arrays. The goal is to use the available sample to reconstruct the levels of the activator $\eta(t)$, which is unobserved, and the gene profiles via the estimation of the parameters in (5.21). We assume an equal variance for all the genes. For each gene, we work with the average of the two available time series. We model the activator $\eta$ using a basis of cubic splines with equally spaced nodes, that is $\eta(t) = \sum_{k=1}^{15} a_i \phi_i(t)$ where the $\phi_i$'s are elements of the basis and $a_1, \ldots, a_{15}$ are parameters to estimate. We apply the methodology described in 5.3 and 5.4. We select the optimal regularization parameter $\lambda$ by using the AIC as model selection criterion.

Figure 5.6 a) shows the estimated profile for the gene SCO3235, which well fits the observed data. The reconstructed gene profiles exhibit a similar fit for the remaining genes. The reconstructed cdaR activator is shown in Figure 5.6 b) together with two independent replicate profiles obtained from a different experiment and not used in the estimation process. The values are normalized between 0 and 1 since the activity of the cdaR protein is expressed in arbitrary units and can be interpreted as relative levels. The estimated profile fits the observed data showing two hills around time points 4 and 9, similarly to the genes profiles. This agrees with the fact that cdaR is an activator of the genes activity. These results show the ability of the proposed approach to identify unknown elements in the ODEs systems by estimation of the parameters of the model.

## 5.7 Summary

We have proposed a new method to estimate general systems of ordinary differential equations measured with noise. Our proposal is based on penalization of the log-likelihood of the problem by means of the ODEs. A reproducing kernel Hilbert space approach has provided the theoretical framework to make this idea feasible. The

concept of Green's function and the connection between linear differential operators and kernels have been used to rewrite the penalized log-likelihood of the problem in a more practical manner.

The main merit of the method is its ability to perform in a single step the estimation problem without solving the system of differential equations. An actual example from system biology has been used to illustrate the utility of this method in scenarios with hidden components.

## 5.A Proof of Proposition 1

We prove the proposition by showing that $l_\lambda(\boldsymbol{\theta}, \boldsymbol{\beta}|\boldsymbol{y}(\boldsymbol{t}), \boldsymbol{x}^\star(\boldsymbol{t}))$ and $g_\lambda(\boldsymbol{\theta}, \boldsymbol{\beta}|\boldsymbol{y}(\boldsymbol{t}), \boldsymbol{x}^\star(\boldsymbol{t}))$ are equal. Using (5.16) and properties of RKHS yields

$$\begin{aligned} l_\lambda(\boldsymbol{\theta}, \boldsymbol{\beta}|\boldsymbol{y}(\boldsymbol{t}), \boldsymbol{x}^\star(\boldsymbol{t})) &= -\sum_{k=1}^{d} \frac{1}{2\sigma_k^2} \|\boldsymbol{y}_k - \boldsymbol{x}_k\|^2 - \frac{\lambda}{2} \sum_{k=1}^{d} \|\mathbf{P}_{\theta_k} \boldsymbol{x}_k - \boldsymbol{f}_k^\star\|^2 \\ &= -\sum_{k=1}^{d} \frac{1}{2\sigma_k^2} \|\widetilde{\boldsymbol{y}}_k - \widetilde{\boldsymbol{x}}_k\|^2 - \frac{\lambda}{2} \sum_{k=1}^{d} \|\mathbf{P}_{\theta_k} \widetilde{\boldsymbol{x}}_k\|^2 \\ &= -\sum_{k=1}^{d} \frac{1}{2\sigma_k^2} \|\widetilde{\boldsymbol{y}}_k - \mathbf{K}_{\theta_k} \boldsymbol{\alpha}_k\|^2 - \frac{\lambda}{2} \sum_{k=1}^{d} \boldsymbol{\alpha}_k^\top \mathbf{K}_{\theta_k} \boldsymbol{\alpha}_k. \end{aligned}$$

Writing the norm in terms of the inner product yields

$$l_\lambda(\boldsymbol{\theta}, \boldsymbol{\beta}|\boldsymbol{y}(\boldsymbol{t}), \boldsymbol{x}^\star(\boldsymbol{t})) = -\sum_{k=1}^{d} \frac{1}{2\sigma_k^2} \left( \widetilde{\boldsymbol{y}}_k^\top \widetilde{\boldsymbol{y}}_k - 2\widetilde{\boldsymbol{y}}_k^\top \mathbf{K}_{\theta_k} \boldsymbol{\alpha}_k + \boldsymbol{\alpha}_k^\top \mathbf{K}_{\theta_k}^\top \mathbf{K}_{\theta_k} \boldsymbol{\alpha}_k + \sigma_k^2 \lambda \boldsymbol{\alpha}_k^\top \mathbf{K}_{\theta_k} \boldsymbol{\alpha}_k \right).$$

For fixed $\{\boldsymbol{\theta}, \boldsymbol{\beta}\}$ and $\sigma_k^2$ the maximum of $l_\lambda(\boldsymbol{\theta}, \boldsymbol{\beta}|\boldsymbol{y}(\boldsymbol{t}), \boldsymbol{x}^\star(\boldsymbol{t}))$ is given for the vectors $\boldsymbol{\alpha}_k = (\mathbf{K}_{\theta_k} + \sigma_k^2 \lambda \mathbf{I}_n)^{-1} \widetilde{\boldsymbol{y}}_k$. Substituting each $\boldsymbol{\alpha}_k$ and simplifying the obtained expression we obtain

$$\begin{aligned} l_\lambda(\boldsymbol{\theta}, \boldsymbol{\beta}|\boldsymbol{y}(\boldsymbol{t}), \boldsymbol{x}^\star(\boldsymbol{t})) &= -\sum_{k=1}^{d} \frac{1}{2\sigma_k^2} \Big\{ \widetilde{\boldsymbol{y}}_k^\top \widetilde{\boldsymbol{y}}_k - 2\widetilde{\boldsymbol{y}}_k^\top \mathbf{K}_{\theta_k} (\mathbf{K}_{\theta_k} + \sigma_k^2 \lambda \mathbf{I}_n)^{-1} \widetilde{\boldsymbol{y}}_k \\ &+ \widetilde{\boldsymbol{y}}_k^\top (\mathbf{K}_{\theta_k} + \sigma_k^2 \lambda \mathbf{I}_n)^{-1} (\mathbf{K}_{\theta_k} + \sigma_k^2 \lambda \mathbf{I}_n) \mathbf{K}_{\theta_k} (\mathbf{K}_{\theta_k} + \sigma_k^2 \lambda \mathbf{I}_n)^{-1} \widetilde{\boldsymbol{y}}_k \Big\} \\ &= -\sum_{k=1}^{d} \frac{1}{2\sigma_k^2} \left\{ \widetilde{\boldsymbol{y}}_k^\top \widetilde{\boldsymbol{y}}_k - \widetilde{\boldsymbol{y}}_k^\top \mathbf{K}_{\theta_k} (\mathbf{K}_{\theta_k} + \sigma_k \lambda \mathbf{I}_n)^{-1} \widetilde{\boldsymbol{y}}_k \right\} \\ &= -\sum_{k=1}^{d} \frac{1}{2\sigma_k^2} \widetilde{\boldsymbol{y}}_k^\top \left\{ \mathbf{I}_n - (\mathbf{I}_n + \sigma_k^2 \lambda \mathbf{K}_{\theta_k}^{-1})^{-1} \right\} \widetilde{\boldsymbol{y}}_k \\ &= g_\lambda(\boldsymbol{\theta}, \boldsymbol{\beta}|\boldsymbol{y}(\boldsymbol{t}), \boldsymbol{x}^\star(\boldsymbol{t})) \end{aligned}$$

as we aimed to prove.

## 5.B Derivation of the AIC for the maximum penalized likelihood estimator

To simplify the derivation, we first prove a general result and then we apply it to our case. Assume that we have an observation $\boldsymbol{y} = (y_1, \ldots, y_n)^\top$ from $\mathcal{N}(\boldsymbol{\mu}, \sigma^2 \mathbf{I}_n)$, with the density $f_{\boldsymbol{\mu}}(\boldsymbol{y})$, where $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)^\top$. Let $f_{\mu_i}(y_i)$ be the density function of the distribution $\mathcal{N}(\mu_i, \sigma^2)$, for $i = 1, \ldots, n$ . Independence of $y_1, \ldots, y_n$ implies

$$\log f_{\boldsymbol{\mu}}(\boldsymbol{y}) = \sum_{k=1}^{n} \log f_{\mu_i}(y_i). \tag{5.22}$$

Let the estimator of $\boldsymbol{\mu}$ have the form $\widehat{\boldsymbol{\mu}} = \mathbf{S}\boldsymbol{y}$, i.e. $\widehat{\boldsymbol{\mu}}$ is a linear smoother. The aim is to find an estimate of the prediction error, the so called *in-sample error* denoted by $\widehat{\mathrm{Err}}_{\mathrm{inn}}$ (Hastie et al., 2009), for the predictor $\widehat{\boldsymbol{\mu}}$. To this end we use the following formula

$$\widehat{\mathrm{Err}}_{\mathrm{inn}} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{\mu}_i)^2 + \frac{2}{n}\sum_{k=1}^{n}\mathrm{cov}(\widehat{\mu}_i, y_i)$$

(Hastie et al., 2009, chapter 7). Up to an additive constant we have $-2\sigma^2 \log f_{\widehat{\mu}_i}(y_i) = (y_i - \widehat{\mu}_i)^2$ and consequently

$$\widehat{\mathrm{Err}}_{\mathrm{inn}} \cong -\frac{2\sigma^2}{n}\sum_{i=1}^{n}\log f_{\widehat{\mu}_i}(y_i) + \frac{2}{n}\sum_{k=1}^{n}\mathrm{cov}(\widehat{\mu}_i, y_i). \tag{5.23}$$

Because $\widehat{\boldsymbol{\mu}}$ is a linear smoother it holds that $\sum_{i=1}^{n}\mathrm{cov}(\widehat{\mu}_i, y_i) = \mathrm{tr}(\mathbf{S})\sigma^2$ (Hastie et al., 2009). After substituting the previous equality in (5.23), using equality (5.22) and scaling with $n/\sigma^2$ we obtain that up to a constant

$$\widehat{\mathrm{Err}}_{\mathrm{inn}} = -2\log f_{\boldsymbol{\mu}}(\boldsymbol{y}) + 2\mathrm{tr}(\mathbf{S}).$$

Substituting $l(\widehat{\boldsymbol{\mu}}) = \log f_{\widehat{\boldsymbol{\mu}}}(\boldsymbol{y})$ into the previous equality we obtain the expression denoted by AIC

$$\mathrm{AIC} = -2l(\widehat{\boldsymbol{\mu}}) + 2\mathrm{df}, \tag{5.24}$$

where $\mathrm{df} = \mathrm{tr}(\mathbf{S})$.

Now we apply the derived result to our case. We have that $\widetilde{\boldsymbol{y}}_k \sim \mathcal{N}(\widetilde{\boldsymbol{x}}_k, \sigma_k^2 \mathbf{I}_n)$ and

$\widehat{\widetilde{\boldsymbol{x}}}_k = \widehat{\mathbf{S}}_{\lambda,k}\widetilde{\boldsymbol{y}}_k$, where $\widehat{\mathbf{S}}_{\lambda,k} = \mathbf{K}_{\widehat{\theta}_k}(\mathbf{K}_{\widehat{\theta}_k} + \lambda\widehat{\sigma}_k^2\mathbf{I}_n)^{-1}\widetilde{\boldsymbol{y}}_k$. According to (5.24), the AIC score for the estimate of the $k$th state is

$$\mathrm{AIC}_k(\lambda) = -2l_k(\boldsymbol{\theta}, \boldsymbol{\beta}|\boldsymbol{y}(\boldsymbol{t}), \boldsymbol{x}^\star(\boldsymbol{t})) + 2\mathrm{tr}(\widehat{\mathbf{S}}_{\lambda,k}),$$

where $l_k(\boldsymbol{\theta}, \boldsymbol{\beta}|\boldsymbol{y}(\boldsymbol{t}), \boldsymbol{x}^\star(\boldsymbol{t})) = \frac{1}{2\sigma_k^2}\|\boldsymbol{y}_k - \boldsymbol{x}_k\|^2$. Summing expressions $\mathrm{AIC}_k$, for $k = 1, \ldots, d$, we obtain

$$\mathrm{AIC}(\lambda) = -2l(\boldsymbol{\theta}, \boldsymbol{\beta}|\boldsymbol{y}(\boldsymbol{t}), \boldsymbol{x}^\star(\boldsymbol{t})) + 2\sum_{k=1}^{d}\mathrm{tr}(\widehat{\mathbf{S}}_{\lambda,k}).$$

The derivation presented here is simpler than the one suggested in González et al. (2013), which assumes the measure of prediction error to be the deviance and uses the theory of general covariance penalties (Efron, 2004, 1986).

# Chapter 6

# Inferring latent gene regulatory network kinetics

Regulatory networks consist of gene encoding transcription factors (TFs) and the genes they activate or repress. Various types of systems of ordinary differential equations have been proposed to model these networks, ranging from linear to Michaelis-Menten approaches. In practice, a serious drawback to estimating these models is that the TFs are generally unobserved because of a lack of high-throughput techniques to measure the abundance of proteins in the cell. The challenge is to infer their activity profile together with the kinetic parameters of the ODEs using level expression measurements of the genes they regulate. In this chapter we show how the framework presented in the previous chapter can be used to infer the kinetic parameters of regulatory networks with one or more TFs using time course gene expression data. Our approach is also able to predict the activity levels of the TF. We illustrate this using the example of an SOS repair system in Escherichia coli. The reconstructed TF exhibits a behaviour similar to experimentally measured profiles and the genetic expression data are fitted well.

The remainder of this chapter is divided into four sections. In Section 6.1, we introduce a model for observed gene data expression with one transcription factor. In Section 6.2, we detail the estimation procedure, which for this specific example is complemented by an EM algorithm. In Section 6.3, our statistical framework is applied to 6 time-course gene expression data in the SOS system in *Escherichia coli* with one TF. We show some results regarding the reconstruction of the TF and the fit of the model to the data. The last section contains the summary and the appendix contains proofs.

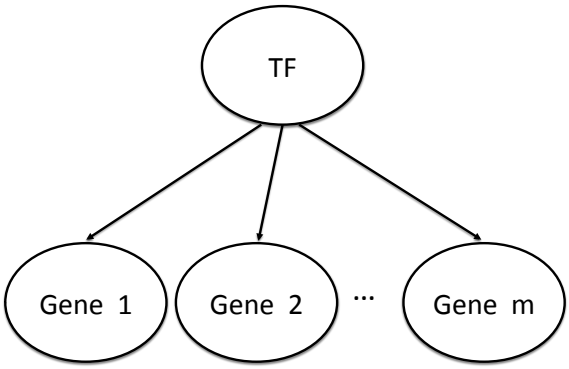



Fig. 6.1 Single Input Motif (SIM) of a gene regulatory network with one transcription factor.

## 6.1 System and methods

### 6.1.1 Modelling transcriptional GRN with ODE models

In gene regulatory networks, the variables of interest are the concentrations of mRNA molecules and the abundance of proteins produced by a set of $d$ genes. To simplify, in the sequel we assume that one gene contains the information to produce only one protein. We denote by $\boldsymbol{\eta}(t) = (\eta_1(t), \ldots, \eta_d(t))^\top$ the abundance of the proteins (TFs) and by $\boldsymbol{x}(t) = (x_1(t), \ldots, x_d(t))^\top$ the concentrations of the mRNA molecules at time $t$. We consider that $t$ varies in some time interval $[0, T]$, in which the GRN is studied. Following mass-action kinetics we assume that the expressions of the genes of the network on average satisfy the ODEs

$$x_k'(t) = p(t; \boldsymbol{\theta}_k, \boldsymbol{\eta}) - \delta_k x_k(t), \tag{6.1}$$

for $k = 1, \ldots, d$, where $\delta_k$s are the degradation rates of mRNAs and $p(t; \boldsymbol{\theta}_k, \boldsymbol{\eta})$ is a function that describes how the TFs regulate the gene $k$ for some parameter vector $\boldsymbol{\theta}_k$. In general, it is assumed that the TFs satisfy

$$\eta_k'(t) = \beta_k^\eta x_k(t) - \delta_k^\eta \eta_k(t),$$

where $\delta_k^\eta$ is the protein degradation rate and $\beta_k^\eta$ is the translational rate for gene $k$.

In the literature several models have been considered to define $p(t; \boldsymbol{\theta}_k, \boldsymbol{\eta})$ in (6.1)

ranging from linear approaches (Chen et al., 1999) to non-parametric methods (Äijö and Lähdesmäki, 2009). In practice, experimental work suggests that the response of the mRNA abundance to the concentration of a TF follows a *Hill curve* (De Jong, 2002). This response can be well described by the Michaelis-Menten (MM) formulation. In case of activation of gene $k$ by the transcription factor $s$, the transcription function is assumed to satisfy

$$p^+(t;\boldsymbol{\theta}_k,\eta_s) = \beta_k \frac{\eta_s(t)}{\gamma_k + \eta_s(t)} + \varphi_k,$$

for $\boldsymbol{\theta}_k = (\varphi_k, \beta_k, \gamma_k)^\top$. Similarly, in cases of repression, the response can be modelled by

$$p^-(t;\boldsymbol{\theta}_k,\eta_s) = \beta_k \frac{1}{\gamma_k + \eta_s(t)} + \varphi_k. \tag{6.2}$$

If a gene is regulated by several TFs, a product of type-$p^+$ and type-$p^-$ functions can be used to model the regulatory component of expression (6.1). The only non-standard part in this model is the presence of $\varphi_k$, which is added to detect possible non-specific activation.

### 6.1.2 GRN with one TF: single input motif

In expression (6.1) several genes encoding TFs are involved in the model. Nevertheless, these types of regulatory networks are difficult to identify given the actual lack of reliable methods for measuring TF activities. For this reason, networks involving only one TF and the genes it regulates as in Figure 6.1 are the most studied in the literature and are the primary focus of our following analysis.

These one-to-many patterns of interaction between one TF and several genes are called single input motifs (SIM), a term first introduced in Milo et al. (2002). Within a SIM, the expression of gene $k$ depends on the decay constant $\delta_k$ and on the transcription function $p(t;\boldsymbol{\theta}_k,\eta)$ where $\eta$ is the unique transcription factor of the network.

The profile $\eta$ is unobserved and has to be reconstructed from the expression of the genes. In this work we assume that $\eta$ is a function which can be written in terms of a basis of spline functions defined in $[0,T]$. That is,

$$\eta(t) = \sum_{j=1}^{m} \mu_j \phi_j(t), \tag{6.3}$$

where $\mu_j \in \mathbb{R}$ and $\{\phi_1,\ldots,\phi_m\}$ is a truncated-power basis set.

Although the transcription factor $\eta$ is common to all target genes in the network, the kinetic parameters of each gene are expected to be target-dependent. That is,

$$x_k(t) = x_k(t; \delta_k, \boldsymbol{\theta}_k, \boldsymbol{\mu}),$$

where $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_m)^\top$ is the vector of weights of the TF in the spline representation (6.3). Biologically, it is reasonable to assume that the gene-specific parameters $\boldsymbol{\theta}_k$ and $\delta_k$ are the same across different biological conditions. However, this is not the case with the initial amount of gene expression $x_k(0)$ since the gene transcription can be affected by an uncontrolled number of external conditions.

### 6.1.3 Noise model

Let $y_{ki}$ denote the measured expression of gene $k$ at a time-point $t_i$. We assume that the observed gene expression measurements of the target genes are conditionally independent given the transcription factor activity, and that each target gene $k$ is normally distributed with location parameter $x_k(t)$ and scale parameter $\sigma_k^2(t)$. That is, we assume that

$$y_{ki} \sim \mathcal{N}(x_k(t_i), \sigma_k^2(t_i)). \tag{6.4}$$

In our context, the log-likelihood contribution of a single observation is, up to an additive constant, given by

$$l(x_k(t_i), \sigma_{ki}^2 | y_{ki}) = -\frac{1}{2}\left\{\frac{y_{ki} - x_k(t_i)}{\sigma_{ki}}\right\}^2 - \frac{1}{2}\log(\sigma_{ki}^2).$$

Let $\mathcal{D}_k = \{(y_{ki}, t_i) \in \mathbb{R} \times [0, T]\}_{i=1}^n$ denote the set of gene expression measurements of gene $k$ across the time points $t_1, \ldots, t_n$. Then the contribution of each gene $k$ to the log-likelihood of the network is

$$l_k(\delta_k, \boldsymbol{\theta}_k, \boldsymbol{\Sigma}_k, \boldsymbol{\mu} | \mathcal{D}_k) = \sum_{i=1}^n l(x_k(t_i), \sigma_{ki}^2 | y_{ki}),$$

for $\boldsymbol{\Sigma}_k = \mathrm{diag}(\sigma_{k1}^2, \ldots, \sigma_{kn}^2)$ and where it is assumed that each function $x_k$ satisfies (6.1).

### 6.1.4 Penalized log-likelihood of a GRN with one TF

The system of differential equations (6.1) describes the dynamics of the gene regulatory system in interval $[0, T]$. However, in practical scenarios, we have access to only a finite number of measurements of the gene expression levels. Following the approach from the previous chapter we approximate the rate of gene expression by the first order difference.

Let $\boldsymbol{t} = (t_1, \ldots, t_n)^\top$ denote the vector whose elements represent the time points in which gene expression measurements are available and define $x_k(\boldsymbol{t}) = (x_k(t_1), \ldots, x_k(t_n))^\top$ and $p(\boldsymbol{t}; \boldsymbol{\theta}_k, \eta) = (p(t_1; \boldsymbol{\theta}_k, \eta), \ldots, p(t_n; \boldsymbol{\theta}_k, \eta))^\top$. Then one can rewrite the discrete version of dynamics of the gene $k$ in (6.1) as

$$\mathbf{D}_n x_k(\boldsymbol{t}) = p(\boldsymbol{t}; \boldsymbol{\theta}_k, \eta) - \delta_k x_k(\boldsymbol{t}),$$

where the matrix $\mathbf{D}_n$ is the difference operator defined in (5.10).

Let $\mathbf{I}_n$ be the identity matrix and denote by $\mathbf{P}_{\delta_k} = \mathbf{D}_n + \delta_k \mathbf{I}_n$ the difference operator associated with gene $k$. To define penalized loss, we assume that $\mathbf{P}_{\delta_k}$ is invertible and introduce

$$\widetilde{x}_k(\boldsymbol{t}) = x_k(\boldsymbol{t}) - \mathbf{P}_{\delta_k}^{-1} p(\boldsymbol{t}; \boldsymbol{\theta}_k, \boldsymbol{\mu}), \tag{6.5}$$

$$\widetilde{\boldsymbol{y}}_k = \boldsymbol{y}_k - \mathbf{P}_{\delta_k}^{-1} p(\boldsymbol{t}; \boldsymbol{\theta}_k, \boldsymbol{\mu}), \tag{6.6}$$

$$\|\widetilde{x}_k(\boldsymbol{t})\|^2_{\mathbf{K}_{\delta_k}} = \boldsymbol{\alpha}_k^\top \mathbf{K}_{\delta_k} \boldsymbol{\alpha}_k,$$

where $\mathbf{K}_{\delta_k} = (\mathbf{P}_{\delta_k}^\top \mathbf{P}_{\delta_k})^{-1}$ and $\boldsymbol{\alpha}_k$ is the vector in $\mathbb{R}^n$ characterizing $\widetilde{x}_k(\boldsymbol{t})$.

Denote by $\widetilde{\mathcal{D}}_k$ the transformed set of expression measurements associated with the gene $k$. Using the framework from the previous chapter, for a GRN where a single TF $\eta$ regulates all genes in the network the penalized log-likelihood can be written as

$$l_\lambda(\Delta, \Theta, \boldsymbol{\Sigma}, \boldsymbol{\mu} | \widetilde{\mathcal{D}}) \;=\; \sum_{k=1}^{d} l_k(\delta_k, \boldsymbol{\theta}_k, \boldsymbol{\Sigma}_k, \boldsymbol{\mu} | \widetilde{\mathcal{D}}_k) - \frac{\lambda}{2} \sum_{k=1}^{d} \|\widetilde{x}_k(\boldsymbol{t})\|^2_{\mathbf{K}_{\delta_k}},$$

where $\widetilde{\mathcal{D}} = \{\widetilde{\mathcal{D}}_1, \ldots, \widetilde{\mathcal{D}}_d\}$ represents the whole sample available for the network; $\Theta$ represents the set of kinetic parameters $\boldsymbol{\theta}_k$, $\Delta = \{\delta_1, \ldots, \delta_d\}$, $\boldsymbol{\Sigma}$ stands for all scale parameters of the normal distribution, and $\boldsymbol{\mu}$ is the set of weights corresponding to the representation of the TF activity in a spline basis.

In the RKHS framework, the (transformed) expression level of each gene $k$ and its

influence matrix are

$$\widetilde{x}_k(\boldsymbol{t}) = \mathbf{K}_{\delta_k}\boldsymbol{\alpha}_k = \mathbf{S}_{\lambda,k}\widetilde{\boldsymbol{y}}_k, \tag{6.7}$$

$$\mathbf{S}_{\lambda,k} = \mathbf{K}_{\delta_k}(\mathbf{K}_{\delta_k} + \lambda\boldsymbol{\Sigma}_k)^{-1}. \tag{6.8}$$

Estimates of the gene expression profiles $x_1, \ldots, x_d$ can be recovered using (6.5) and (6.7).

### 6.1.5 Parameter estimation

Denote by $A = \{\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_d\}$ the set of parameters characterizing the gene profiles $\widetilde{\boldsymbol{x}}$. Then the maximum penalized log-likelihood estimators of $\Delta, \Theta, \boldsymbol{\Sigma}, \boldsymbol{\mu}$ and $A$ are given by

$$(\widehat{\Delta}_\lambda, \widehat{\Theta}_\lambda, \widehat{\boldsymbol{\Sigma}}_\lambda, \widehat{\boldsymbol{\mu}}_\lambda, \widehat{A}_\lambda) = \arg\max_{\Delta,\Theta,\boldsymbol{\Sigma},\boldsymbol{\mu},A} l_\lambda(\Delta, \Theta, \boldsymbol{\Sigma}, \boldsymbol{\mu}, A|\widetilde{\mathcal{D}}). \tag{6.9}$$

### 6.1.6 Model selection

For selecting $\lambda$ we use the AIC, which we have derived in the previous chapter. For the linear smoother (6.7) the influence matrix for gene $k$ is given by (6.8). The effective number of parameters is defined as $\mathrm{df}_k = \mathrm{tr}(\widehat{\mathbf{S}}_{\lambda,k})$. Then the AIC is defined in our context as

$$\mathrm{AIC}_k(\lambda) = -2l_k(\widehat{\delta}_k, \widehat{\boldsymbol{\theta}}_k, \widehat{\boldsymbol{\Sigma}}_k, \widehat{\boldsymbol{\mu}}|\mathcal{D}_k) + 2\mathrm{df}_k.$$

Extending the criterion to all genes of the network, we can select the optimal $\lambda$ as

$$\lambda_{\mathrm{opt}} = \arg\min_\lambda \sum_{k=1}^{d} \mathrm{AIC}_k(\lambda).$$

In the literature, $\lambda$ and $\boldsymbol{\Sigma}_k$ are normally not allowed to change simultaneously. As detailed in Rasmussen (2006), $\lambda$ is sometimes fixed to $1/2$ or to $1/n$ if the variance parameters are allowed to vary. In this work the variance parameters are estimated off-line and the matrices $\boldsymbol{\Sigma}_k$ are fixed in the estimation process. Then the parameter $\lambda$ is obtained using the above mentioned AIC or other sensible model selection criteria.

### 6.1.7 Confidence intervals

In maximum likelihood estimation approaches the most standard way of obtaining the variance of a parameter estimate $\widehat{\boldsymbol{\theta}}$ is by means of the negative Hessian evaluated at

$\widehat{\boldsymbol{\theta}}$. In our context, however, the parameter estimates of the gene regulatory network are obtained by a penalized likelihood approach and the aforementioned approach is not directly applicable.

To circumvent this issue we propose a parametric bootstrap (Efron, 1979) to obtain confidence intervals in our setting. To this aim and by virtue of (6.4) we consider the fitted model $\mathcal{N}(\widehat{x}_k(t_i), \widehat{\sigma_k}^2(t_i))$, where the hat-expressions are estimated values. Then from this model we generate $B$ bootstrap data sets for which we use the proposed method to obtain new estimates of the parameters. The quantiles of the empirical distribution of the bootstrap estimates are then used to estimate the confidence intervals.

## 6.2 Algorithm

In this section we detail the steps to obtain the estimators $\widehat{\Delta}_\lambda$, $\widehat{\Theta}_\lambda$, $\widehat{\boldsymbol{\Sigma}}_\lambda$ and $\widehat{\boldsymbol{\mu}}_\lambda$ in (6.9). Also, estimators for the gene profiles $\widehat{x}_k$ are provided. To this aim, we propose an augmented formulation of the problem in order to provide a way to deal with the common lack of observations in real applications. Since the derivative approximation is better for smaller discretization steps, increasing the number of data points results in a better derivative approximation. To use this idea we propose an EM algorithm to accommodate hidden observations.

### 6.2.1 Augmented data formulation

Denote by $\mathcal{D}_k^O = \{(y_{ki}^O, t_i^O) \in \mathbb{R} \times [0,T]\}_{i=1}^n$ the set of expression measurements of the gene $k$ across the observed time points $t_1^O, \ldots, t_n^O$. Consider a refinement of hidden observations, all different from those in $\mathcal{D}_k^O$, given by $\mathcal{D}^H = \{(y_{ki}^H, t_{ki}^H) \in \mathbb{R} \times [0,T]\}_{i=1}^r$. Denote by

$$\begin{aligned} \boldsymbol{t}^O &= (t_1^O, \ldots, t_n^O)^\top, \, \boldsymbol{t}^H = (t_1^H, \ldots, t_r^H)^\top, \\ \boldsymbol{y}_k^O &= (y_{k1}^O, \ldots, y_{kn}^O)^\top, \, \boldsymbol{y}_k^H = (y_{k1}^H, \ldots, y_{kr}^H)^\top, \end{aligned}$$

the vectors of times and data of both samples for gene $k$. Define $\boldsymbol{t}$ to be the vector resulting from the ordered concatenation of $\boldsymbol{t}^O$ and $\boldsymbol{t}^H$, and let $\boldsymbol{y}_k$ be the concatenation of the vectors $\boldsymbol{y}_k^O$ and $\boldsymbol{y}_k^H$ ordered according to the ordering in $\boldsymbol{t}$. Then the log-

likelihood for the augmented data formulation is

$$\begin{aligned} l_\lambda(\cdot|\widetilde{\mathcal{D}}^O, \widetilde{\mathcal{D}}^H) &= -\frac{1}{2}\sum_{k=1}^{d}(\widetilde{\boldsymbol{y}}_k - \mathbf{K}_{\delta_k}\boldsymbol{\alpha}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\widetilde{\boldsymbol{y}}_k - \mathbf{K}_{\delta_k}\boldsymbol{\alpha}_k) \\ &- \frac{1}{2}\sum_{k=1}^{d}\sum_{i=1}^{n+r}\log(\sigma_{ki}^2) - \lambda\sum_{k=1}^{d}\boldsymbol{\alpha}_k^\top \mathbf{K}_{\delta_k}\boldsymbol{\alpha}_k, \end{aligned} \tag{6.10}$$

where $\widetilde{\boldsymbol{y}}_k$ is the vector of transformed data defined in (6.6), $\mathbf{K}_{\delta_k} = (\mathbf{P}_{\delta_k}^\top \mathbf{P}_{\delta_k})^{-1}$, and $\boldsymbol{\alpha}_k \in \mathbb{R}^{n+r}$.

### 6.2.2 E-step

In the expectation (E) step, we need to calculate the expectation of $l_\lambda(\cdot|\widetilde{\mathcal{D}}_O, \widetilde{\mathcal{D}}_H)$ in terms of the hidden observations. For simplicity of notation, define

$$Q_\lambda(\Delta, \Theta, \boldsymbol{\Sigma}, \boldsymbol{\mu}, A) = E_{\boldsymbol{y}^H}\left\{l_\lambda(\cdot|\widetilde{\mathcal{D}}_O, \widetilde{\mathcal{D}}_H)|\mathcal{D}_O, \Delta^*, \Theta^*, \boldsymbol{\Sigma}^*, \boldsymbol{\mu}^*, A^*\right\}.$$

Consider the matrix $\mathbf{C}_H \in \mathbb{R}^{r\times(n+r)}$ defined by

$$(\mathbf{C}_H)_{ij} = \begin{cases} 1 & \text{if } \boldsymbol{t}_{H_i} = \boldsymbol{t}_j \\ 0 & \text{otherwise} \end{cases}.$$

Then it can be proven that

$$Q_\lambda(\Delta, \Theta, \boldsymbol{\Sigma}, \boldsymbol{\mu}, A) = l_\lambda(\cdot|\widetilde{\mathcal{D}}^O, \widetilde{\mathcal{D}}^{H*}) - \frac{1}{2}\sum_{k=1}^{d}\sum_{i=1}^{r}(\sigma_{ki}^*)^2,$$

for $\widetilde{\mathcal{D}}^{H*} = \{\widetilde{\mathcal{D}}_1^{H*}, \ldots, \widetilde{\mathcal{D}}_d^{H*}\}$, where the augmented vector $\widetilde{\boldsymbol{y}}_k^*$ of observations associated with each gene $k$ is given by the ordered concatenation of the vectors $\widetilde{\boldsymbol{y}}_k^O$ and $\mathbf{C}_H\mathbf{K}_{\delta_k^*}\boldsymbol{\alpha}_k^*$ according to the ordering in $\boldsymbol{t}$. See appendix for details.

### 6.2.3 M-step

In the maximization (M) step, we maximize the augmented log-likelihood over the parameters of interest. Assume that some values for $\Delta^*, \Theta^*, \boldsymbol{\Sigma}^*, \boldsymbol{\mu}^*$ and $A^*$ are given and consider the augmented vectors of observations $\widetilde{\boldsymbol{y}}_k^*$. Then in this step we search for the values of the parameters such that

$$(\widehat{\Delta}, \widehat{\Theta}, \widehat{\boldsymbol{\Sigma}}, \widehat{\boldsymbol{\mu}}, \widehat{A}) = \arg\max_{\Delta, \Theta, \boldsymbol{\Sigma}, \boldsymbol{\mu}} Q_\lambda(\Delta, \Theta, \boldsymbol{\Sigma}, \boldsymbol{\mu}, A).$$

In practice, the problem can be split into two different maximization problems. Parameters $\Delta, \Theta, \mathbf{\Sigma}$ and $\boldsymbol{\mu}$ can be calculated independently of $A$ by taking

$$\begin{aligned}(\widehat{\Delta}, \widehat{\Theta}, \widehat{\mathbf{\Sigma}}, \widehat{\boldsymbol{\mu}}) &= \arg \max_{\Delta,\Theta,\mathbf{\Sigma},\boldsymbol{\mu}} -\frac{1}{2}\sum_{k=1}^{d}(\widetilde{\boldsymbol{y}}_k^*)^\top \mathbf{\Sigma}_k^{-1}(\mathbf{I}-\mathbf{S}_{\lambda,k})\widetilde{\boldsymbol{y}}_k^* \\ &- \frac{1}{2}\sum_{k=1}^{d}\sum_{i=1}^{n+r}\log(\sigma_{ki}^2) - \frac{1}{2}\sum_{k=1}^{d}\sum_{i=1}^{r}(\sigma_{ki}^*)^2, \end{aligned} \tag{6.11}$$

where $\mathbf{S}_{\lambda,k}$ is the influence matrix associated with each gene defined in (6.8). Regarding the set of parameters $A$, made up of the vectors $\boldsymbol{\alpha}_1, \ldots . \boldsymbol{\alpha}_d$, it can be shown using standard methods of differential calculus that they can be calculated by

$$\widehat{\boldsymbol{\alpha}}_k = (\mathbf{K}_{\widehat{\delta}_k} + \lambda\widehat{\mathbf{\Sigma}}_k)^{-1}\widetilde{\boldsymbol{y}}_k^*. \tag{6.12}$$

See appendix for details.

### 6.2.4 EM algorithm

In order to apply the EM algorithm we need to consider some initial values of the parameters. If no other information about the system is provided, a reasonable way to proceed is to smooth each $\widetilde{\boldsymbol{y}}_k^O$ with some standard smoothing technique. In this way for each individual gene we obtain a reasonable initial set of hidden observations $\widetilde{\boldsymbol{y}}_k^H$. Then we can define $\widetilde{\boldsymbol{y}}_k$ as the ordered concatenation of the vectors $\widetilde{\boldsymbol{y}}_k^O$ and $\widetilde{\boldsymbol{y}}_k^H$ according to the ordering in $\boldsymbol{t}$. Assuming some values for $\Delta^*, \Theta^*, \mathbf{\Sigma}^*$ and $\boldsymbol{\mu}^*$ we can now calculate each $\boldsymbol{\alpha}_k^* = (\widetilde{\mathbf{K}}_{\delta_k^*} + \lambda\mathbf{\Sigma}_k^*)^{-1}\widetilde{\boldsymbol{y}}_k$ to obtain reasonable initial value for $A^*$. The steps of the EM algorithm are detailed next.

1. **Start**: Consider some initial values $\Delta^*, \Theta^*, \mathbf{\Sigma}^*, \boldsymbol{\mu}^*$ and $A^*$.
2. **E-step**: Compute each $\mathbf{K}_{\delta_k^*}$ and obtain $\widetilde{\boldsymbol{y}}_k^*$.
3. **M-step**: Obtain $(\widehat{\Delta}, \widehat{\Theta}, \widehat{\mathbf{\Sigma}}, \widehat{\boldsymbol{\mu}})$ as in (6.11) using a conjugate gradient algorithm and each $\widehat{\boldsymbol{\alpha}}_k$ using (6.12). Go to step 2.
4. **End**: Upon convergence, take $(\widehat{\Delta}_\lambda, \widehat{\Theta}_\lambda, \widehat{\mathbf{\Sigma}}_\lambda, \widehat{\boldsymbol{\mu}}_\lambda) = (\widehat{\Delta}, \widehat{\Theta}, \widehat{\mathbf{\Sigma}}, \widehat{\boldsymbol{\mu}})$ and $\widehat{\boldsymbol{\alpha}}_{\lambda,k} = \widehat{\boldsymbol{\alpha}}_k$ for $k = 1 \ldots, d$.

## 6.3 SOS repair system in Escherichia coli

Changes in the expression levels of genes in *Escherichia coli* as a result of DNA damage (SOS response) have been extensively studied in the last few years. However their

behaviour has not been completely understood (Khanin et al., 2006). The SOS system includes more than 30 genes controlled by a (transcriptional repressor) protein called LexA. Under normal conditions the levels of LexA are high and the expression of the gene is repressed. With DNA damage the LexA protein is inactivated, causing the up-regulation of genes suppressed by the LexA under normal conditions. The aim of this analysis is twofold: first, to reconstruct the activity level of the repressor LexA by using the expression profiles of its target genes, and second, to identify the system and to order the genes of the SOS system in terms of the speed with which they are repressed by the LexA protein. To this aim we use the proposed penalized likelihood approach.

### 6.3.1 The data set and the goal

The data set used for this experiment is made up of 14 expression genes (dinF, dinI, lexA, recA, recN, ruvA, ruvB, sbmC, sulA, umuC, umuD, uvrB, yegG and ijW) of the Escherichia coli SOS system. These 14 genes are targets of the master repressor LexA and their expression is studied under UV exposure (40 $J/m^2$) in both wild-type cells and lexA1 mutants. (Khanin et al., 2006). The abundance of the mRNA molecules associated with the genes was measured at six time points: 0, 5, 10, 20, 40 and 60 minutes. Raw data are normalized as detailed in (Khanin et al., 2006). The master repressor is unobserved. Following expression (6.3) we assume that its activity can be described by a cubic spline function with $d = 5$ basis functions. In this example we assume a Michaelis-Menten formulation (see 6.2). The goal of this experiment is to use this gene expression sample to reconstruct the activity of the repressor $\eta(t)$ and to estimate the kinetic parameters in (6.1). In addition, the gene profiles will be inferred by means of (6.5) and (6.7). The parameters to be estimated are the kinetic parameters of the 14 genes together with the TF weights and the vectors $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_{14}$ characterizing the gene profiles.

### 6.3.2 Estimation process

Only six data points are available per gene. This lack of observations could cause instabilities in the estimation process which we deal with as follows. First, the variance parameters are assumed to be constant for the data within the same gene. We estimate $\sigma_1^2, \ldots, \sigma_{14}^2$ off-line by fitting smoothing splines to the data and estimating the residual variance in each case. The TF factor is assumed to be normalized between zero and

one, which robustifies the estimation of the parameters of the system. In practice this is not a problem since the repression or activation of the LexA protein is expressed in arbitrary units, which can be interpreted as relative levels.

Reconstruction of the TF and estimation of the gene-specific kinetic parameters are done in two steps. In both, maximization of the penalized log-likelihood is achieved by means of the conjugate gradient method. First we estimate the TF profile. To do so we estimate the system without intermediate points. The penalization parameter $\lambda$ is calculated by using the AIC as detailed in Section 6.1.6. We consider the estimates of the weights of the spline basis $\widehat{\mu}_1, \ldots, \widehat{\mu}_5$ and we reconstruct the TF profile. We observe that the values of the $\widehat{\varphi}_i$ are zero for most of the genes. Given the lack of data, in order to gain more precision in the estimation of the remaining kinetic parameters we assume all the $\widehat{\varphi}_i$ to be zero and we re-estimate each gene independently considering the TF profile obtained above to be fixed. A different penalization term is now recalculated for each gene using the AIC. Four intermediate points are considered in order to improve the estimation. Confidence intervals for the parameters were obtained using a parametric bootstrap as detailed in Section 6.1.7.

### 6.3.3 Reconstruction of the LexA activity

The reconstructed LexA repressor is shown in Figure 6.2. The smoothed LexA profile, obtained using a cubic spline, is shown. The crucial aspect of the obtained profile is that it agrees with the behaviour of experimentally observed profiles in Ronen et al. (2002) and Sassanfar and Roberts (1990). It has been observed that after irradiation the amount of LexA decreases, and after a recovery phase it increases again. This is the behaviour shown in Figure 6.2 where the level of the LexA decreases to zero within the first 20 minutes to completely recover by 60 minutes.

### 6.3.4 Inferred kinetics profiles

The reconstructed gene profiles show a good fit with the data in the 14 cases. In Figure 6.3 we show the data and the estimated profiles for the genes dinI and recN. These two genes were selected because they exhibit different types of profiles. Gene dinI shows a fast increase in regulation up until min 20-30, and later descends gradually. This coincides with the time point in which the master repressor starts to recover. On the other hand gene recN is stable in time after minute 20. The introduced intermediate points help to recover a smooth version of the gene profiles.

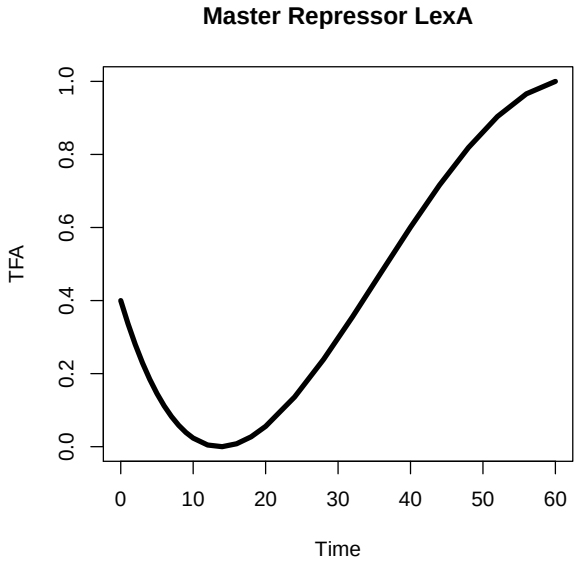



Fig. 6.2 Reconstruction of the activity of the master repressor LexA scaled between 0 and 1. The smoothed LexA profile is obtained using a cubic spline. Time is given in minutes.

### 6.3.5 Estimated kinetic parameters and interpretation

The values of the estimated gene-dependent kinetic parameters and their corresponding 95% confidence intervals are shown in Table 6.1. Two genes recN and umuC, show significant differences in parameters when compared to the rest. These two genes are the only ones in the database that do not show a decrease in expression pattern after minute 20. This seems to indicate a misspecification in the model for these two genes. Particularly, as also suggested in (Khanin et al., 2006), this type of behaviour can be modelled by a linear degradation ODE $\dot{x} = \varphi + \delta x$.

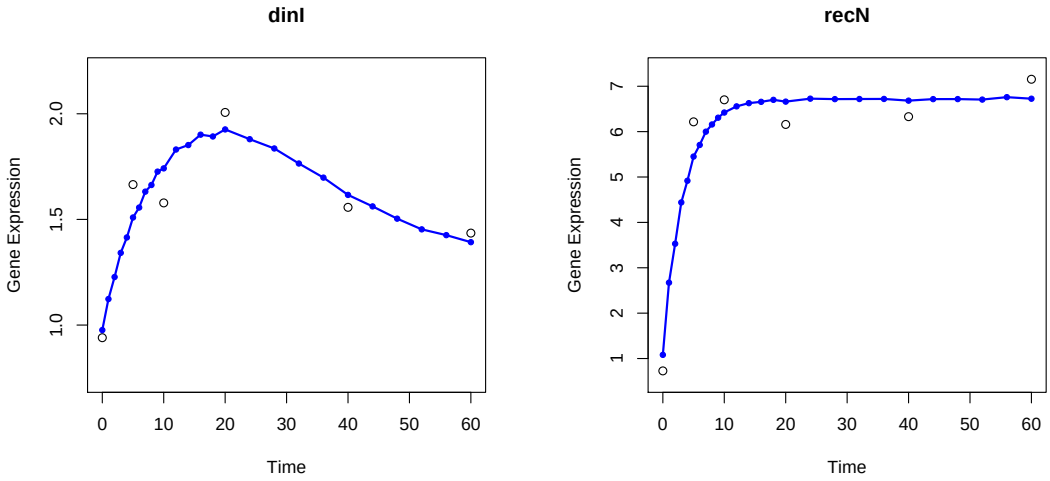
The remaining genes are sorted in the table by using the effective production rate calculated by

$$r_k = \frac{\beta_k}{\gamma_k + \bar{\eta}} \times 100,$$

where $\bar{\eta}$ is the average TF level. Genes ijW ruvA and lexA are the quickest to be regulated. Genes sbmC dinF are the slowest. Regarding the precision of the estimates we observe that confidence intervals for the parameters in genes ijW and dinF are larger than for the rest of the genes, which indicates that for these genes the data are probably too noisy.

| Gene | $\sigma_k$ | $r_k$ | $\widehat{\beta}_k$ | $CI_{95\%}(\beta_k)$ | $\widehat{\delta}_k$ | $CI_{95\%}(\delta_k)$ | $\widehat{\gamma}_k$ | $CI_{95\%}(\gamma_k)$ |
|---|---|---|---|---|---|---|---|---|
| recN | 0.37 | - | 6912.05 | (6465.54, 6933.12) | 0.29 | (0.21, 0.30) | 3568.39 | (3527.46, 4433.54) |
| umuC | 0.38 | - | 79.94 | (78.22, 88.71) | 0.09 | (0.09, 0.10) | 123.21 | (117.45, 124.12) |
| ijW | 0.04 | 0.80 | 12.44 | (2.90, 28.20) | 0.42 | (0.21, 0.46) | 22.54 | (9.05, 76.17) |
| ruvA | 0.06 | 0.34 | 5.27 | (3.82, 6.30) | 0.42 | (0.35, 0.45) | 6.02 | (4.64, 7.15) |
| lexA | 0.07 | 0.31 | 4.75 | (2.28, 7.56) | 0.33 | (0.25, 0.38) | 7.70 | (4.23, 11.75) |
| sulA | 0.55 | 0.28 | 4.36 | (1.03, 9.24) | 0.39 | (0.15, 0.40) | 2.66 | (1.42, 6.88) |
| umuD | 0.08 | 0.19 | 2.99 | (2.13, 3.94) | 0.15 | (0.13, 0.16) | 5.11 | (4.08, 6.66) |
| yegG | 0.04 | 0.14 | 2.10 | (1.79, 2.42) | 0.23 | (0.21, 0.25) | 3.72 | (3.31, 4.20) |
| ruvB | 0.01 | 0.13 | 1.98 | (1.79, 2.23) | 0.27 | (0.26, 0.29) | 4.38 | (4.11, 4.74) |
| uvrB | 0.04 | 0.12 | 1.89 | (1.26, 3.40) | 0.10 | (0.09, 0.11) | 7.11 | (5.07, 12.08) |
| dinI | 0.08 | 0.06 | 0.99 | (0.56, 1.39) | 0.21 | (0.15, 0.27) | 2.36 | (1.71, 2.83) |
| recA | 0.19 | 0.03 | 0.48 | (0.39, 0.67) | 0.11 | (0.10, 0.13) | 0.85 | (0.72, 1.11) |
| sbmC | 0.04 | 0.02 | 0.26 | (0.22, 0.30) | 0.11 | (0.11, 0.12) | 0.71 | (0.62, 0.78) |
| dinF | 0.02 | 0.00 | 5.01 | (2.06, 32.16) | 0.12 | (0.08, 0.18) | 31.59 | (17.32, 167.07) |

Table 6.1 Parameter estimates and confidence intervals for the 14 genes of the Ecoli-SOS system. Above, genes recN and umuC whose expression does not decline after minute 20. Below, the 12 remaining genes of the database which decline after minute 20 sorted by the ratio $r_k$. 95 % confidence intervals are calculated using a parametric bootstrap.




(a) Estimated profile of the gene dinI. This gene exhibits an up-regulation after UV radiation. Its expression levels decline after minute 20.

(b) Estimated profile of the gene recN. This gene exhibits an up-regulation after UV radiation. Its expression levels remain stable after minute 20.

Fig. 6.3 Data and reconstructed profiles of two genes which represent the two expression patterns found in the database. Raw data are represented by empty points. Dense points represent the values of the estimated profiles in the 6 observed and 20 hidden points of each gene.

## 6.4 Summary

In this chapter we have presented an application of the RKHS method from the previous chapter in order to infer GRN with one hidden TF from time-course expression measurements. The proposed approach does not require the transcription factor activity to have a predefined shape, and a general spline representation allows it to capture the dynamics of the TF. The EM algorithm has been proposed to estimate the model due to the lack of observed data points.

The proposed method was applied in the reconstruction of the SOS repair system in Escherichia Coli. In this example, the reconstructed TF exhibits a behaviour similar to that seen in (independent) experimentally measured profiles. In addition the gene expression data are fitted and the results are coherent with those obtained in previous works (Khanin et al., 2006).

## 6.A Notation

For the proofs below we introduce some notation. We define a matrix $\mathbf{C}_O \in \mathbb{R}^{n\times(n+r)}$ such that

$$(\mathbf{C}_O)_{ij} = \begin{cases} 1 & \text{if } \boldsymbol{t}_{O_i} = \boldsymbol{t}_j \\ 0 & \text{otherwise} \end{cases},$$

vectors $\Phi_k = (\delta_k, \boldsymbol{\theta}_k, \boldsymbol{\Sigma}_k, \boldsymbol{\mu}, \boldsymbol{\alpha}_k)$, $\Phi_k^* = (\delta_k^*, \boldsymbol{\theta}_k^*, \boldsymbol{\Sigma}_k^*, \boldsymbol{\mu}^*, \boldsymbol{\alpha}_k^*)$ and functions

$$\begin{aligned} l_{\lambda,k}(\Phi_k) &= l_k(\Phi_k|\widetilde{\mathcal{D}}_k) - \frac{\lambda}{2}\boldsymbol{\alpha}_k^\top \mathbf{K}_{\delta_k}\boldsymbol{\alpha}_k \\ &= -\frac{1}{2}(\widetilde{\boldsymbol{y}}_k - \mathbf{K}_{\delta_k}\boldsymbol{\alpha}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\widetilde{\boldsymbol{y}}_k - \mathbf{K}_{\delta_k}\boldsymbol{\alpha}_k) \\ &- \frac{1}{2}\sum_{i=1}^{n+r}\log(\sigma_{ki}^2) - \frac{\lambda}{2}\boldsymbol{\alpha}_k^\top \mathbf{K}_{\delta_k}\boldsymbol{\alpha}_k, \end{aligned}$$

$k = 1, \ldots, d$. With this notation we can write the penalized log-likelihood as

$$l_\lambda(\Delta, \Theta, \boldsymbol{\Sigma}, \boldsymbol{\mu}) = \sum_{k=1}^{d} l_{\lambda,k}(\Phi_k).$$

## 6.B Proof of the expectation step of the EM algorithm

*Proof.* For every $k = 1, \ldots, d$ we calculate $E_{\boldsymbol{y}_k^H}\left(l_{\lambda,k}|\mathcal{D}_k^O, \Phi_k^*\right)$. Denote with $\boldsymbol{\Sigma}_{H,k}$ and $\boldsymbol{\Sigma}_{O,k}$ diagonal matrices whose diagonals are vectors of variances of observed and hidden observations of the $k$th equation respectively. Splitting the log-likelihood into two parts corresponding to the hidden and the observed observations we obtain that

$$\begin{aligned} E_{\boldsymbol{y}_k^H}\left(l_{\lambda,k}|\mathcal{D}_k^O, \Phi_k^*\right) &= -\frac{1}{2}\sum_{i=1}^{n+r}\log\sigma_{ki}^2 - \frac{\lambda}{2}\boldsymbol{\alpha}_k^\top \mathbf{K}_{\delta_k}\boldsymbol{\alpha}_k - \frac{1}{2}\|\boldsymbol{y}_k^O - \mathbf{C}_O\mathbf{K}_{\delta_k}\boldsymbol{\alpha}_k\|^2_{(\boldsymbol{\Sigma}_{O,k}^*)^{-1}} \\ &- \frac{1}{2}E_{\boldsymbol{y}_k^H}\left(\|\boldsymbol{y}_k^H - \mathbf{C}_H\mathbf{K}_{\delta_k}\boldsymbol{\alpha}_k\|^2_{(\boldsymbol{\Sigma}_{H,k}^*)^{-1}}|\mathcal{D}_k^O, \delta_k^*, \boldsymbol{\Sigma}_k^*, \boldsymbol{\alpha}_k^*\right), \end{aligned}$$

where $\|\boldsymbol{x}\|_{\mathbf{A}}^2 = \boldsymbol{x}^\top\mathbf{A}\boldsymbol{x}$. In the previous expression we have that

$$\begin{aligned} E_{\boldsymbol{y}_k^H}\left(\|\boldsymbol{y}_k^H - \mathbf{C}_H\mathbf{K}_{\delta_k}\boldsymbol{\alpha}_k\|^2_{\boldsymbol{\Sigma}_k}|\mathcal{D}_k^O, \Phi_k^*\right) &= E_{\boldsymbol{y}_k^H}\left\{(\boldsymbol{y}_k^H)^\top(\boldsymbol{\Sigma}_{H,k}^*)^{-1}\boldsymbol{y}_k^H|\mathcal{D}_k^O, \delta_k^*, \boldsymbol{\Sigma}_k^*, \boldsymbol{\alpha}_k^*\right\} \\ &- 2E_{\boldsymbol{y}_k^H}\left\{(\boldsymbol{y}_k^H)^\top(\boldsymbol{\Sigma}_{H,k}^*)^{-1}\mathbf{C}_H\mathbf{K}_{\delta_k}\boldsymbol{\alpha}_k|\mathcal{D}_k^O, \delta_k^*, \boldsymbol{\Sigma}_k^*, \boldsymbol{\alpha}_k^*\right\} \\ &+ \boldsymbol{\alpha}_k^\top\mathbf{K}_{\delta_k}\mathbf{C}_H^\top(\boldsymbol{\Sigma}_{H,k}^*)^{-1}\mathbf{C}_H\mathbf{K}_{\delta_k}\boldsymbol{\alpha}_k. \end{aligned}$$

By using the properties of the expectation and the variance and by factorizing terms we obtain that

$$E_{\boldsymbol{y}_k^H}\left(\|\boldsymbol{y}_k^H - \mathbf{C}_H\mathbf{K}_{\delta_k}\boldsymbol{\alpha}_k\|^2_{\boldsymbol{\Sigma}_k}|\mathcal{D}_k^O,\Phi_k^*\right) = \|\mathbf{C}_H\mathbf{K}_{\delta_k^*}\boldsymbol{\alpha}_k^* - \mathbf{C}_H\mathbf{K}_{\delta_k}\boldsymbol{\alpha}_k\|^2_{(\boldsymbol{\Sigma}_{H,k}^*)^{-1}} + \sum_{i=1}^{r}(\sigma_{ki}^*)^2.$$

Substituting the last expression into the expected log-likelihood we obtain that

$$\begin{aligned} E_{\boldsymbol{y}_k^H}\left(l_{\lambda,k}|\mathcal{D}_k^O,\Phi_k^*\right) &= -\frac{1}{2}\sum_{i=1}^{n+r}\log\sigma_{ki}^2 - \frac{\lambda}{2}\boldsymbol{\alpha}_k^\top\mathbf{K}_{\delta_k}\boldsymbol{\alpha}_k - \frac{1}{2}\|\boldsymbol{y}_k^O - \mathbf{C}_O\mathbf{K}_{\delta_k}\boldsymbol{\alpha}_k\|^2_{(\boldsymbol{\Sigma}_{O,k}^*)^{-1}} \\ &- \frac{1}{2}\left\{\|\mathbf{C}_H\mathbf{K}_{\delta_k^*}\boldsymbol{\alpha}_k^* - \mathbf{C}_H\mathbf{K}_{\delta_k}\boldsymbol{\alpha}_k\|^2_{(\boldsymbol{\Sigma}_{H,k}^*)^{-1}} + \sum_{i=1}^{r}(\sigma_{ki}^*)^2\right\}. \end{aligned}$$

Define $\boldsymbol{y}_k^* = (\boldsymbol{y}_k^O, \mathbf{C}_H\mathbf{K}_{\delta_k^*}\boldsymbol{\alpha}_k^*)^\top$. Grouping the terms we obtain that

$$\begin{aligned} E_{\boldsymbol{y}_k^H}\left(l_{\lambda,k}(\boldsymbol{y}_k^O,\boldsymbol{y}_k^H|\Phi_k)|\mathcal{D}_k^O,\Phi_k^*\right) &= -\frac{1}{2}\sum_{i=1}^{n+r}\log\sigma_{ki}^2 - \frac{1}{2}\|\widetilde{\boldsymbol{y}}_k^* - \mathbf{K}_{\delta_k}\boldsymbol{\alpha}_k\|^2_{\boldsymbol{\Sigma}_k^{-1}} \\ &- \frac{\lambda}{2}\boldsymbol{\alpha}_k^\top\mathbf{K}_{\delta_k}\boldsymbol{\alpha}_k - \frac{1}{2}\sum_{i=1}^{r}(\sigma_{ki}^*)^2. \end{aligned}$$

The proof is completed by summing over $k$'s and taking into account (6.10).

□

# 6.C Proof of the maximization step of the EM algorithm

*Proof.* Denote by $\boldsymbol{y}_k^* = (\boldsymbol{y}_k^O, \mathbf{C}_H\mathbf{K}_{\delta_k^*}\boldsymbol{\alpha}_k^*)^\top$. Then for fixed $\delta_k$ and $\boldsymbol{\Sigma}_k$ the maximum of $E_{\boldsymbol{y}_k^H}\left(l_{\lambda,k}|\mathcal{D}_k^O,\delta_k^*,\boldsymbol{\Sigma}_k^*,\boldsymbol{\alpha}_k^*\right)$ is given for the vector $\boldsymbol{\alpha}_k = (\mathbf{K}_{\delta_k} + \lambda\boldsymbol{\Sigma}_k)^{-1}\boldsymbol{y}_k^*$. By substituting $\boldsymbol{\alpha}_k$ into the expression and simplifying we obtain that

$$E_{\boldsymbol{y}_k^H}\left(l_{\lambda,k}|\mathcal{D}_k^O,\Phi_k^*\right) = -\frac{1}{2}\sum_{i=1}^{n+r}\log\sigma_{ki}^2 - \frac{1}{2}\sum_{i=1}^{r}(\sigma_{ki}^*)^2 - \frac{1}{2}(\widetilde{\boldsymbol{y}}_k^*)^\top\boldsymbol{\Sigma}_k^{-1}(\mathbf{I} - \mathbf{S}_{\lambda,k})\widetilde{\boldsymbol{y}}_k^*,$$

where $\mathbf{S}_{\lambda,k} = \mathbf{K}_{\delta_k}(\mathbf{K}_{\delta_k} + \lambda\boldsymbol{\Sigma}_k^2)^{-1}$. By taking sum over $k$'s we obtain

$$\begin{aligned} (\widehat{\Delta},\widehat{\Theta},\widehat{\boldsymbol{\Sigma}},\widehat{\boldsymbol{\mu}}) &= \arg\max_{\Delta,\Theta,\boldsymbol{\Sigma},\boldsymbol{\mu}} -\frac{1}{2}\sum_{k=1}^{d}(\widetilde{\boldsymbol{y}}_k^*)^\top\boldsymbol{\Sigma}_k^{-1}(\mathbf{I} - \mathbf{S}_{\lambda,k})\widetilde{\boldsymbol{y}}_k^* \\ &- \frac{1}{2}\sum_{k=1}^{d}\sum_{i=1}^{n+r}\log(\sigma_{ki}^2) - \frac{1}{2}\sum_{k=1}^{m}\sum_{i=1}^{r}(\sigma_{ki}^*)^2, \end{aligned}$$

as we aimed to prove. □

# Conclusion

In Chapter 1, we have displayed Table 1.1 and Table 1.2. The former lists existing methods for choosing the regularization parameter in Gaussian graphical models; the latter contains those for estimating parameters in ordinary differential equations. The results presented in this thesis complement both tables.

We have complemented Table 1.1 with two new methods, the generalized information criterion and the Kullback-Leibler cross-validation. These methods improve the Akaike's information criterion and generalized approximate cross-validation. The purpose of Table 1.1 is not only to give a summary of the methods and show our contribution, but more importantly, to make a clear distinction between different methods. This distinction with respect to prediction and model selection does not seem to be stressed enough in the literature. For example, usage of cross-validation (CV) for model selection is not uncommon (Wang and Pillai, 2013; Gao et al., 2012) even though it was pointed out that this method is not appropriate for that purpose (Liu et al., 2010; Lian, 2011). However, to our best knowledge, so far it has not been actually proved that cross-validation is not model selection consistent. Therefore, as a step in that direction, we have proved that this claim is true for the Akaike's information criterion (AIC). Since AIC and CV share similar asymptotic properties (Shao, 1997; Yang, 2005) in the regression setting, it is plausible that the same holds for Gaussian graphical models. Moreover, we expect the same result to hold for other methods whose aim is prediction. Thus we conclude and conjecture the following:

- AIC, $K$-CV, GACV, GIC and KLCV are appropriate for prediction but not for model selection.
- BIC, EBIC, StARS are appropriate for model selection.

Another important issue is the generality of the methods. Although the methods we propose have advantages, they rely on the assumption of Gaussianity. In regard to this, computational methods $K$-CV and StARS are expected to be the most robust with respect to the departure from Gaussianity.

As for Table 1.2, we have both proposed a new method and extended an existing one. The one that we propose, the RKHS approach, can deal with systems of differential equations of a general form. The method is relatively simple to use and produces estimates of the parameters in one step. It has been tested on real and simulated data and it exhibits reasonable performance. However, the drawback is that it relies on the assumption of Gaussianity and its asymptotic properties are unclear. It seems likely that the method is not asymptotically consistent, due to the various approximations we employ.

On the other hand, although the second method, the window-based estimator, can be applied only to autonomous systems of differential equations linear in the parameters, it does provide an asymptotically consistent estimator. Although we restrict the class of systems we consider, many applied problems are defined by the systems that belong to this class. The window-based estimator has an explicit form, it is computationally fast and it does not require strong assumptions on the distribution of the data.

Finally, both methods discussed can be applied only if all states of the system are observed. Estimation of the parameters in the partially observed systems has not been covered in this thesis. In the literature two ways are used to deal with partially observed systems. One way is to use a Bayesian approach which relies on Gaussian processes (Chkrebtii et al., 2013). The other involves writing the unobserved states as a linear combination of certain basis functions and optimizing with respect to both the parameters of the system and the coefficients of the basis expansion (Ramsay et al., 2007). In the case of the window estimator both approaches can be used; this will be the topic of our further research.

# Summary

In this thesis we treat the problem of inference of two types of models: Gaussian graphical models and ordinary differential equation models.

After presenting the background material in Chapter 2, we address the problem of estimating the precision matrix in Gaussian graphical models. Every precision matrix corresponds to a certain conditional independence graph. The main idea which we we stress is that one can obtain an estimate of a precision matrix that yields a good model in terms of prediction but whose corresponding conditional independence graph is poor. We prove that Akaike's information criterion is not model selection consistent in a fixed $p$ setting. We propose two criteria that yield predictively accurate models and we show how these criteria can be used to obtain models with a good conditional independence graph; this is the content of Chapter 3.

In Chapter 4, we extend to the time course data the existing method for estimating autonomous systems of differential equations linear in the parameters, developed for repeated measurements. The linearity of the system in parameters allows one to explicitly obtain the estimators of the parameters without solving the differential equation. This is of great importance since no optimization is needed. The method does not require any initial values and is extremely fast. Furthermore, the obtained estimator is asymptotically consistent.

In Chapter 5, we develop a method for estimating parameters in general systems of differential equations. We dicretize the system of differential equations and use the reproducing kernel Hilbert space to define an approximation of the log-likelihood and its corresponding regularizer. By optimizing the obtained approximated penalized log-likelihood we estimate the parameters without solving the differential equation.

Finally, in the last chapter we apply the method developed in Chapter 5 to inference of gene regulatory kinetics. One ingredient present here and not in Chapter 5 is the use of the EM algorithm which we employed to deal with the lack of observations.

# Samenvatting

In dit proefschrift bestuderen we het probleem van inferentie voor twee typen statistische modellen, te weten Gaussische grafische modellen, en modellen die op gewone differentiaalvergelijkingen zijn gebaseerd.

Na een beschrijving van de achtergrond in hoofdstuk 2, bekijken we in hoofdstuk 3 het probleem van het schatten van de concentratiematrix van Gaussische grafische modellen. Elke concentratiematrix correspondeert met een bepaalde voorwaardelijke onafhankelijkheidsgraaf. We benadrukken het feit dat het mogelijk is om een benadering van de concentratiematrix te verkrijgen die een model met een goede voorspellende waarde oplevert, maar waarvan de voorwaardelijke onafhankelijkheidsgraaf erg onnauwkeurig is. We bewijzen bovendien dat Akaikes informatiecriterium voor constante $p$ niet leidt tot consistente modelselectie. Er worden twee nieuwe criteria voorgesteld die leiden tot modellen die nauwkeurige voorspellingen opleveren en we laten ook zien hoe deze criteria gebruikt kunnen worden om modellen met goede voorwaardelijke onafhankelijkheidsgrafen te verkrijgen.

In hoofdstuk 4 breiden we de bestaande methode voor het schatten van autonome stelsels differentiaalvergelijkingen met lineaire parameters uit naar het geval waarin er herhaalde metingen in de tijd worden uitgevoerd. Doordat de parameters van het stelsel lineair zijn, is het mogelijk om een expliciete uitdrukking te vinden voor de schatters van de parameters, zonder dat het stelsel differentiaalvergelijkingen opgelost hoeft te worden. Dit is van groot belang, omdat hiervoor geen optimalisatieproces nodig is. De zo gevonden methode heeft geen beginvoorwaarden nodig en is zeer efficient. Bovendien is de verkregen schatter asymptotisch consistent.

In hoofdstuk 5 ontwikkelen we een methode voor het schatten van parameters van algemene stelsels differentiaalvergelijkingen. We discretizeren het stelsel en gebruiken een Hilbertruimte met reproducerende kern om een benadering van de waarschijnlijkheidsfunctie en de bijbehorende regularisator te definiëren. Door het optimaliseren van de verkregen benadering van de waarschijnlijkheidsfunctie waarbij een boeteterm inbegrepen is, schatten we de parameters van het stelsel differentiaalvergelijkingen,

zonder dit stelsel op te hoeven lossen.

In het laatste hoofdstuk passen we de in hoofdstuk 5 ontwikkelde methode toe op inferentie van de reactiekinetiek van genregulatienetwerken. Een nieuw onderdeel in dit hoofdstuk is het gebruik van het EM-algoritme, toegepast om om te gaan met een beperkte aantal waarnemingen in de tijd .

# References


Abbruzzo, A., Vujačić, I., Wit, E., and Mineo, A. M. (2014). Generalized information criterion for model selection in penalized graphical models. *arXiv preprint arXiv:1403.1249*.

Äijö, T. and Lähdesmäki, H. (2009). Learning gene regulatory networks from gene expression measurements using non-parametric molecular kinetics. *Bioinformatics*, 25(22):2937–2944.

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In *Second international symposium on information theory*, pages 267–281. Akademinai Kiado.

Baldi, P., Brunak, S., Chauvin, Y., Andersen, C. A., and Nielsen, H. (2000). Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, 16(5):412–424.

Banerjee, O., El Ghaoui, L., and d'Aspremont, A. (2008). Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *The Journal of Machine Learning Research*, 9:485–516.

Bard, Y. (1974). *Nonlinear parameter estimation*. Academic Press New York.

Barenco, M., Tomescu, D., Brewer, D., Callard, R., Stark, J., and Hubank, M. (2006). Ranked prediction of p53 targets using hidden variable dynamic modeling. *Genome biology*, 7(3):R25.

Barton, G. (1989). *Elements of Green's functions and propagation: potentials, diffusion, and waves*. Oxford University Press.

Bates, D. M. and Watts, D. G. (1988). *Nonlinear regression: iterative estimation and linear approximations*. Wiley.

Bellman, R. and Roth, R. S. (1971). The use of splines with unknown end points in the identification of systems. *Journal of Mathematical Analysis and Applications*, 34(1):26–33.

Bernstein, D. S. (2009). *Matrix mathematics: theory, facts, and formulas*. Princeton University Press.

Bickel, P. J. and Ritov, Y. (2003). Nonparametric estimators which can be "plugged-in". *The Annals of Statistics*, 31(4):1033–1053.

Biegler, L., Damiano, J., and Blau, G. (1986). Nonlinear parameter estimation: a case study comparison. *AIChE Journal*, 32(1):29–45.

Bock, H. G. (1983). *Recent advances in parameter identification techniques for O.D.E.* Springer.

Bonhoeffer, S., May, R. M., Shaw, G. M., and Nowak, M. A. (1997). Virus dynamics and drug therapy. *Proceedings of the National Academy of Sciences*, 94(13):6971–6976.

Breiman, L. (1996). Heuristics of instability and stabilization in model selection. *The Annals of Statistics*, 24(6):2350–2383.

Brunel, N. J. B. (2008). Parameter estimation of ode's via nonparametric estimators. *Electronic Journal of Statistics*, 2:1242–1267.

Bühlmann, P. and Van De Geer, S. (2011). *Statistics for high-dimensional data: methods, theory and applications*. Springer.

Cai, T., Liu, W., and Luo, X. (2011). A constrained $l_1$ minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, 106(494):594–607.

Calderhead, B., Girolami, M., and Lawrence, N. (2008). Accelerating bayesian inference over nonlinear differential equations with gaussian processes. *Advances in Neural Information Processing*, 21:217–224.

Campbell, D. and Steele, R. J. (2012). Smooth functional tempering for nonlinear differential equation models. *Statistics and Computing*, 22(2):429–443.

Campbell, D. A. (2007). *Bayesian collocation tempering and generalized profiling for estimation of parameters from differential equation models*. PhD thesis, McGill University, Montreal, Quebec.

Cao, J., Fussmann, G. F., and Ramsay, J. O. (2008). Estimating a predator-prey dynamical model with the parameter cascades method. *Biometrics*, 64(3):959–967.

Cao, J. and Ramsay, J. O. (2007). Parameter cascades and profiling in functional data analysis. *Computational Statistics*, 22(3):335–351.

Cao, J. and Zhao, H. (2008). Estimating dynamic models for gene regulation networks. *Bioinformatics*, 24(14):1619–1624.

Chen, J. and Wu, H. (2008a). Efficient local estimation for time-varying coefficients in deterministic dynamic models with applications to hiv-1 dynamics. *Journal of the American Statistical Association*, 103(481):369–384.

Chen, J. and Wu, H. (2008b). Estimation of time-varying parameters in deterministic dynamic models. *Statistica Sinica*, 18(3):987–1006.

Chen, T., He, H. L., Church, G. M., et al. (1999). Modeling gene expression with differential equations. In *Pacific symposium on biocomputing*, volume 4, pages 29–40.

Chen, X. (2012). *Lasso-type sparse regression and high-dimensional Gaussian graphical models*. PhD thesis, The University of British Columbia, Vancouver, Canada.

Chen, Z. and Haykin, S. (2002). On different facets of regularization theory. *Neural Computation*, 14(12):2791–2846.

Chkrebtii, O., Campbell, D. A., Girolami, M. A., and Calderhead, B. (2013). Bayesian uncertainty quantification for differential equations. *arXiv preprint arXiv:1306.2365*.

Dattner, I. and Klaassen, C. A. (2013). Estimation in systems of ordinary differential equations linear in the parameters. *arXiv preprint arXiv:1305.4126*.

De Jong, H. (2002). Modeling and simulation of genetic regulatory systems: a literature review. *Journal of computational biology*, 9(1):67–103.

Dempster, A. P. (1972). Covariance selection. *Biometrics*, 28(1):157–175.

Donnet, S. and Samson, A. (2007). Estimation of parameters in incomplete data models defined by dynamical systems. *Journal of Statistical Planning and Inference*, 137(9):2815–2831.

Duffy, D. (2001). *Green's functions with applications*. CRC Press.

Earn, D. J., Rohani, P., Bolker, B. M., and Grenfell, B. T. (2000). A simple model for complex dynamical transitions in epidemics. *Science*, 287(5453):667–670.

Edelstein-Keshet, L. (2005). *Mathematical models in biology*, volume 46. Siam.

Edwards, D. (2000). *Introduction to graphical modelling*. Springer.

Efron, B. (1979). Bootstrap methods: another look at the jackknife. *The Annals of Statistics*, 7(1):1–26.

Efron, B. (1986). How biased is the apparent error rate of a prediction rule? *Journal of the American Statistical Association*, 81(394):461–470.

Efron, B. (2004). The estimation of prediction error: Covariance penalties and cross-validation. *Journal of the American Statistical Association*, 99(467):619–642.

Ellner, S. P., Seifu, Y., and Smith, R. H. (2002). Fitting population dynamic models to time-series data by gradient matching. *Ecology*, 83(8):2256–2270.

Fan, J., Feng, Y., and Wu, Y. (2009). Network exploration via the adaptive lasso and scad penalties. *The Annals of Applied Statistics*, 3(2):521–541.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360.

Fang, Y., Wu, H., and Zhu, L.-X. (2011). A two-stage estimation method for random coefficient differential equation models with application to longitudinal hiv dynamic data. *Statistica Sinica*, 21(3):1145–1170.

Fasshauer, G. E. (2012). Green's functions: Taking another look at kernel approximation, radial basis functions, and splines. In *Approximation Theory XIII: San Antonio 2010*, pages 37–63. Springer.

Fellinghauer, B., Bühlmann, P., Ryffel, M., Von Rhein, M., and Reinhardt, J. D. (2013). Stable graphical model estimation with random forests for discrete, continuous, and mixed variables. *Computational Statistics & Data Analysis*, 64:132–152.

Fine, P. E. and Clarkson, J. A. (1982). Measles in england and wales—i: an analysis of factors underlying seasonal patterns. *International journal of epidemiology*, 11(1):5–14.

Finkenstädt, B. F. and Grenfell, B. T. (2000). Time series modelling of childhood diseases: a dynamical systems approach. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 49(2):187–205.

Fitch, A. M. (2012). *Computationally tractable fitting of graphical models: the cost and benefits of decomposable Bayesian and penalized likelihood approaches*. PhD thesis, Massey University, Albany, New Zealand.

FitzHugh, R. (1955). Mathematical models of threshold phenomena in the nerve membrane. *The bulletin of mathematical biophysics*, 17(4):257–278.

FitzHugh, R. (1961). Impulses and physiological states in theoretical models of nerve membrane. *Biophysical journal*, 1(6):445–466.

Folland, G. B. (1992). *Fourier analysis and its applications*. Wadsworth & Brooks/Cole.

Foygel, R. and Drton, M. (2010). Extended bayesian information criteria for gaussian graphical models. *Advances in Neural Information Processing Systems*, 23:604–612.

Fried, R. and Vogel, D. (2010). On robust gaussian graphical modelling. In *Recent Developments in Applied Probability and Statistics*. Berlin, Heidelberg: Springer.

Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441.

Gao, X., Pu, D. Q., Wu, Y., and Xu, H. (2012). Tuning parameter selection for penalized likelihood estimation of gaussian graphical model. *Statistica Sinica*, 22(3):1123.

Gelman, A., Bois, F., and Jiang, J. (1996). Physiological pharmacokinetic analysis using population modeling and informative prior distributions. *Journal of the American Statistical Association*, 91(436):1400–1412.

Girolami, M. (2008). Bayesian inference for differential equations. *Theoretical Computer Science*, 408(1):4–16.

Goldstein, L. and Messer, K. (1992). Optimal plug-in estimators for nonparametric functional estimation. *The Annals of Statistics*, 20(3):1306–1328.

González, J., Vujačić, I., and Wit, E. (2013). Inferring latent gene regulatory network kinetics. *Statistical applications in genetics and molecular biology*, 12(1):109–127.

González, J., Vujačić, I., and Wit, E. (2014). Reproducing kernel hilbert space based estimation of systems of ordinary differential equations. *Pattern Recognition Letters*, 45:26–32.

Griffel, D. H. (1985). *Applied functional analysis*. Ellis Horwood Limited.

Gugushvili, S. and Klaassen, C. A. J. (2012). $\sqrt{n}$-consistent parameter estimation for systems of ordinary differential equations: bypassing numerical integration via smoothing. *Bernoulli*, 18(3):1061–1098.

Gugushvili, S. and Spreij, P. (2012). Parametric inference for stochastic differential equations: a smooth and match approach. *Latin American Journal of Probability and Mathematical Statistics*, 9(2):609–635.

Hall, P. and Ma, Y. (2014). Quick and easy one-step parameter estimation in differential equations. *To be published in Journal of the Royal Statistical Society: Series B (Statistical Methodology)*.

Harville, D. A. (2008). *Matrix algebra from a statistician's perspective*. Springer, 2 edition.

Hastie, T., Tibshirani, R., and Friedman, J. J. H. (2009). *The elements of statistical learning*. Springer New York, 2 edition.

He, D., Ionides, E. L., and King, A. A. (2010). Plug-and-play inference for disease dynamics: measles in large and small populations as a case study. *Journal of the Royal Society Interface*, 7(43):271–283.

Hemker, P. (1972). Numerical methods for differential equations in system simulation and in parameter estimation. *Analysis and Simulation of biochemical systems*, pages 59–80.

Himmelblau, D., Jones, C., and Bischoff, K. (1967). Determination of rate constants for complex kinetics models. *Industrial & Engineering Chemistry Fundamentals*, 6(4):539–543.

Hooker, G. (2007). Theorems and calculations for smoothing-based profiled estimation of differential equations. Technical report, Technical Report BU-1671-M, Dept. Bio. Stat. and Comp. Bio., Cornell University.

Hooker, G. (2009). Forcing function diagnostics for nonlinear dynamics. *Biometrics*, 65(3):928–936.

Hooker, G. and Biegler, L. (2007). Ipopt and neural dynamics: Tips, tricks and diagnostics. Technical report, Department of Biological Statistics and Computational Biology, Cornell University.

Hooker, G., Ellner, S. P., Roditi, L. D. V., and Earn, D. J. (2011). Parameterizing state–space models for infectious disease dynamics by generalized profiling: measles in ontario. *Journal of The Royal Society Interface*, 8(60):961–974.

Hooker, G., Xiao, L., and Ramsay, J. (2012). Collocinfer: collocation inference for dynamic systems. *R package version 0.1.7*.

Huang, Y., Liu, D., and Wu, H. (2006). Hierarchical bayesian methods for estimation of parameters in a longitudinal hiv dynamic system. *Biometrics*, 62(2):413–423.

Huang, Y. and Wu, H. (2006). A bayesian approach for estimating antiviral efficacy in hiv dynamic models. *Journal of Applied Statistics*, 33(2):155–174.

Huppert, A., Barnea, O., Katriel, G., Yaari, R., Roll, U., and Stone, L. (2012). Modeling and statistical analysis of the spatio-temporal patterns of seasonal influenza in israel. *PloS one*, 7(10):e45107.

Jost, C. and Ellner, S. P. (2000). Testing for predator dependence in predator-prey dynamics: a non-parametric approach. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 267(1453):1611–1620.

Kelley, W. G. and Peterson, A. C. (2010). *The Theory of Differential Equations: Classical and Qualitative*. Springer, 2 edition.

Khanin, R., Vinciotti, V., Mersinias, V., Smith, C., and Wit, E. (2007). Statistical reconstruction of transcription factor activity using michaelis–menten kinetics. *Biometrics*, 63(3):816–823.

Khanin, R., Vinciotti, V., and Wit, E. (2006). Reconstructing repressor protein levels from expression of gene targets in escherichia coli. *Proceedings of the National Academy of Sciences*, 103(49):18592–18596.

Konishi, S. and Kitagawa, G. (1996). Generalised information criteria in model selection. *Biometrika*, 83(4):875–890.

Konishi, S. and Kitagawa, G. (2008). *Information criteria and statistical modeling*. Springer.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.

Lalam, N. and Klaassen, C. A. (2006). *Pseudo maximum likelihood estimation for differential equations*. Eurandom.

Lam, C. and Fan, J. (2009). Sparsistency and rates of convergence in large covariance matrix estimation. *The Annals of Statistics*, 37(6B):4254–4278.

Lauritzen, S. L. (1996). *Graphical models*. Oxford University Press.

Lawrence, N., Rattray, M., Honkela, A., and Titsias, M. (2011). Gaussian process inference for differential equation models of transcriptional regulation. *HandBook of Statistical Systems Biology*, pages 376–394.

Lawrence, N. D., Sanguinetti, G., and Rattray, M. (2006). Modelling transcriptional regulation using gaussian processes. *Advances in Neural Information Processing*, 19:785–792.

Li, H. and Gui, J. (2006). Gradient directed regularization for sparse gaussian concentration graphs, with applications to inference of genetic networks. *Biostatistics*, 7(2):302–317.

Li, L., Brown, M. B., Lee, K.-H., and Gupta, S. (2002). Estimation and inference for a spline-enhanced population pharmacokinetic model. *Biometrics*, 58(3):601–611.

Li, Z., Osborne, M. R., and Prvan, T. (2005). Parameter estimation of ordinary differential equations. *IMA Journal of Numerical Analysis*, 25(2):264–285.

Lian, H. (2011). Shrinkage tuning parameter selection in precision matrices estimation. *Journal of Statistical Planning and Inference*, 141(8):2839–2848.

Liang, H. and Wu, H. (2008). Parameter estimation for differential equation models using a framework of measurement error in regression models. *Journal of the American Statistical Association*, 103(484):1570–1583.

Lillacci, G. and Khammash, M. (2010). Parameter estimation and model selection in computational biology. *PLoS computational biology*, 6(3):e1000696.

Liu, H., Han, F., Yuan, M., Lafferty, J., and Wasserman, L. (2012). High-dimensional semiparametric gaussian copula graphical models. *The Annals of Statistics*, 40(4):2293–2326.

Liu, H., Roeder, K., and Wasserman, L. (2010). Stability approach to regularization selection (stars) for high dimensional graphical models. *Advances in Neural Information Processing Systems*, 23:1432–1440.

Loader, C. (1999). *Local Regression and Likelihood*. Springer.

Lotka, A. J. (1910). Contribution to the theory of periodic reactions. *The Journal of Physical Chemistry*, 14(3):271–274.

Lu, T., Liang, H., Li, H., and Wu, H. (2011). High-dimensional odes coupled with mixed-effects modeling techniques for dynamic gene regulatory network identification. *Journal of the American Statistical Association*, 106(496).

Madár, J., Abonyi, J., Roubos, H., and Szeifert, F. (2003). Incorporating prior knowledge in a cubic spline approximation-application to the identification of reaction kinetic models. *Industrial & engineering chemistry research*, 42(17):4043–4049.

Magnus, J. R. and Neudecker, H. (1985). Matrix differential calculus with applications to simple, hadamard, and kronecker products. *Journal of Mathematical Psychology*, 29(4):474–492.

Magnus, J. R. and Neudecker, H. (2007). *Matrix differential calculus with applications in statistics and econometrics*. Wiley & Sons, third edition.

Meinshausen, N. and Bühlmann, P. (2006). High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34(3):1436–1462.

Menéndez, P., Kourmpetis, Y. A., ter Braak, C. J., and van Eeuwijk, F. A. (2010). Gene regulatory networks from multifactorial perturbations using graphical lasso: application to the dream4 challenge. *PloS one*, 5(12):e14147.

Miao, H., Dykes, C., Demeter, L. M., Cavenaugh, J., Park, S. Y., Perelson, A. S., and Wu, H. (2008). Modeling and estimation of kinetic parameters and replicative fitness of hiv-1 from flow-cytometry-based growth competition experiments. *Bulletin of mathematical biology*, 70(6):1749–1771.

Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., and Alon, U. (2002). Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824–827.

Nagumo, J., Arimoto, S., and Yoshizawa, S. (1962). An active pulse transmission line simulating nerve axon. *Proceedings of the IRE*, 50(10):2061–2070.

Nikol'skij, N. K. (1992). *Functional analysis I: linear functional analysis*, volume 19. Springer.

Nishii, R. (1984). Asymptotic properties of criteria for selection of variables in multiple regression. *The Annals of Statistics*, 12(2):758–765.

Ogunnaike, B. A. and Ray, W. H. (1994). *Process dynamics, modeling, and control*. Oxford University Press.

Olinky, R., Huppert, A., and Stone, L. (2008). Seasonal dynamics and thresholds governing recurrent epidemics. *Journal of mathematical biology*, 56(6):827–839.

Pascual, M. and Ellner, S. P. (2000). Linking ecological patterns to environmental forcing via nonlinear time series models. *Ecology*, 81(10):2767–2780.

Penny, W. D. (2001). Kullback-liebler divergences of normal, gamma, dirichlet and wishart densities. Technical report, Wellcome Department of Cognitive Neurology.

Powers, D. (2011). Evaluation: From precision, recall and f-measure to roc., informedness, markedness & correlation. *Journal of Machine Learning Technologies*, 2(1):37–63.

Poyton, A., Varziri, M. S., McAuley, K. B., McLellan, P., and Ramsay, J. O. (2006). Parameter estimation in continuous-time dynamic models using principal differential analysis. *Computers & chemical engineering*, 30(4):698–708.

Putter, H., Heisterkamp, S., Lange, J., and De Wolf, F. (2002). A bayesian approach to parameter estimation in hiv dynamical models. *Statistics in medicine*, 21(15):2199–2214.

Qi, X. and Zhao, H. (2010). Asymptotic efficiency and finite-sample properties of the generalized profiling estimation of parameters in ordinary differential equations. *The Annals of Statistics*, 38(1):435–481.

Ramsay, J. and B.W., S. (2006). *Functional data analysis*. Springer, 2 edition.

Ramsay, J. O. (1996). Principal differential analysis: Data reduction by differential operators. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(3):495–508.

Ramsay, J. O., Hooker, G., Campbell, D., and Cao, J. (2007). Parameter estimation for differential equations: a generalized smoothing approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(5):741–796.

Rasmussen, C. E. (2006). *Gaussian processes for machine learning*. The MIT Press.

Ravikumar, P., Wainwright, M. J., Raskutti, G., and Yu, B. (2011). High-dimensional covariance estimation by minimizing $l_1$-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935–980.

Roach, G. F. (1982). *Green's functions*. Cambridge University Press, 2 edition.

Rogers, S., Khanin, R., and Girolami, M. (2007). Bayesian model-based inference of transcription factor activity. *BMC bioinformatics*, 8(Suppl 2):S2.

Ronen, M., Rosenberg, R., Shraiman, B. I., and Alon, U. (2002). Assigning numbers to the arrows: parameterizing a gene regulation network by using accurate expression kinetics. *Proceedings of the national academy of sciences*, 99(16):10555–10560.

Rothman, A. J., Bickel, P. J., Levina, E., and Zhu, J. (2008). Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics*, 2:494–515.

Rue, H. and Held, L. (2005). *Gaussian Markov random fields: theory and applications*. CRC Press.

Sassanfar, M. and Roberts, J. W. (1990). Nature of the sos-inducing signal in *Escherichia coli*: The involvement of dna replication. *Journal of molecular biology*, 212(1):79–96.

Schmidt, M. (2010). *Graphical model structure learning with $l_1$-regularization*. PhD thesis, The University of British Columbia, Vancouver, Canada.

Secrier, M., Toni, T., and Stumpf, M. P. (2009). The abc of reverse engineering biological signalling systems. *Molecular Biosystems*, 5(12):1925–1935.

Shao, J. (1997). An asymptotic theory for linear model selection. *Statistica Sinica*, 7(2):221–242.

Smola, A. J. et al. (2002). *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. MIT press.

Stakgold, I. and Holst, M. J. (1979). *Green's functions and boundary value problems*. John Wiley & Sons.

Steinke, F. and Schölkopf, B. (2008). Kernels, regularization and differential equations. *Pattern Recognition*, 41(11):3271–3286.

Steinwart, I. and Christmann, A. (2008). *Support vector machines*. Springer.

Stone, L., Olinky, R., and Huppert, A. (2007). Seasonal dynamics of recurrent epidemics. *Nature*, 446(7135):533–536.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):111–147.

Stone, M. (1977). An asymptotic equivalence of choice of model by cross-validation and akaike's criterion. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):44–47.

Stortelder, W. J. (1996). Parameter estimation in dynamic systems. *Mathematics and Computers in Simulation*, 42(2):135–142.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288.

Tjoa, I. B. and Biegler, L. T. (1991). Simultaneous solution and optimization strategies for parameter estimation of differential-algebraic equation systems. *Industrial & Engineering Chemistry Research*, 30(2):376–385.

Turkington, D. A. (2005). *Matrix calculus and zero-one matrices: Statistical and econometric applications*. Cambridge University Press.

Vajda, S., Valko, P., and Yermakova, A. (1986). A direct-indirect procedure for estimation of kinetic parameters. *Computers & chemical engineering*, 10(1):49–58.

Varah, J. (1982). A spline least squares method for numerical parameter estimation in differential equations. *SIAM Journal on Scientific and Statistical Computing*, 3(1):28–46.

Varziri, M., Poyton, A., McAuley, K., McLellan, P., and Ramsay, J. (2008). Selecting optimal weighting factors in ipda for parameter estimation in continuous-time dynamic models. *Computers & Chemical Engineering*, 32(12):3011–3022.

Voit, E. O. and Savageau, M. A. (1982). Power-low approach to modeling biological systems: Ii. application to ethanol production. *Journal of fermentation technology*, 60(3):229–232.

Voorman, A., Shojaie, A., and Witten, D. (2013). Graph estimation with joint additive models. *Biometrika*, page ast053.

Vujačić, I., Abbruzzo, A., and Wit, E. (2014a). A computationally fast alternative to cross-validation in penalized gaussian graphical models. *arXiv preprint arXiv:1309.6216*.

Vujačić, I., Dattner, I., González, J., , and Wit, E. (2014b). Time-course window estimator for ordinary differential equations linear in the parameters. *manuscript under revision in Statistics and Computing*.

Wainwright, M. J. and Jordan, M. I. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305.

Wang, H. and Pillai, N. S. (2013). On a class of shrinkage priors for covariance matrix estimation. *Journal of Computational and Graphical Statistics*, 22(3):689–707.

Whittaker, J. (2009). *Graphical models in applied multivariate statistics*. Wiley Publishing.

Wild, G. and Seber, C. (1989). *Nonlinear Regression*. Wiley.

Wood, N. S. and Lindgren, F. (2013). Apts statistical computing. Notes.

Xiang, D. and Wahba, G. (1996). A generalized approximate cross validation for smoothing splines with non-gaussian data. *Statistica Sinica*, 6:675–692.

Xue, H., Miao, H., and Wu, H. (2010). Sieve estimation of constant and time-varying coefficients in nonlinear ordinary differential equation models by considering both numerical error and measurement error. *The Annals of Statistics*, 38(4):2351–2387.

Xun, X., Cao, J., Mallick, B., Maity, A., and Carroll, R. J. (2013). Parameter estimation of partial differential equation models. *Journal of the American Statistical Association*, 108(503):1009–1020.

Yanagihara, H., Tonda, T., and Matsumoto, C. (2006). Bias correction of cross-validation criterion based on kullback–leibler information under a general condition. *Journal of multivariate analysis*, 97(9):1965–1975.

Yang, Y. (2005). Can the strengths of aic and bic be shared? a conflict between model indentification and regression estimation. *Biometrika*, 92(4):937–950.

Yuan, M. and Lin, Y. (2007). Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19–35.

Zhang, Y., Li, R., and Tsai, C.-L. (2010). Regularization parameter selections via generalized information criterion. *Journal of the American Statistical Association*, 105(489):312–323.

Zhao, T., Liu, H., Roeder, K., Lafferty, J., and Wasserman, L. (2014). *huge: High-dimensional Undirected Graph Estimation*. R package version 1.2.6.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101(476):1418–1429.

# Curriculum vitae

**Ivan Vujačić** was born on March 6, 1983 in Podgorica, Montenegro. He graduated from the University of Montenegro, Podgorica, Montenegro, with diploma in Theoretical Mathematics in 2007. During the last year of his studies he taught mathematics at the Electrical Engineering High school Vaso Aligrudić, Podgorica. He received the Master of Science degree (MSc) in Mathematics in 2009. From 2007 to 2010 he taught at the Faculty of Mathematics and Natural Sciences, University of Montenegro. In 2010 he began his PhD project in statistics at the Johann Bernoulli Institute for Mathematics and Computer Science at the University of Groningen, The Netherlands. During his PhD he worked on inference of Gaussian graphical models and ordinary differential equations. The results of his research are described in this thesis.

# Inference of Gaussian Graphical models and ordinary differential equations

---

**Ivan Vujačić**
ivanvujacic@gmail.com

Please email me comments and corrections.

ISBN 978-90-367-7098-9 (printed version)
ISBN 978-90-367-7097-2 (electronic version)

Colophon

This thesis was completed using the PhD/MPhil thesis LaTeX template, by Krishna Kumar, University of Cambridge.
*Proofreader*: JoAnn van Seventer (joannvanseventer@home.nl)
*Printed by*: Grafimedia (grafimedia@rug.nl)